

分布式缓存服务

API 参考

文档版本 01
发布日期 2025-09-08



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目 录

1 使用前必读.....	1
2 API 概览.....	3
3 如何调用 API.....	4
3.1 构造请求.....	4
3.2 认证鉴权.....	6
3.3 返回结果.....	8
4 应用示例.....	10
4.1 创建 DCS 高频场景示例.....	10
4.2 示例 1：创建 Redis 实例.....	16
4.3 示例 2：创建数据迁移任务.....	21
4.4 示例 3：批量删除 Redis 实例.....	24
5 API V2（推荐）.....	28
5.1 生命周期管理.....	28
5.1.1 创建缓存实例.....	28
5.1.2 查询所有实例列表.....	70
5.1.3 批量删除实例.....	92
5.1.4 查询指定实例.....	99
5.1.5 删除实例.....	117
5.1.6 修改实例信息.....	123
5.1.7 变更实例规格.....	135
5.1.8 获取可删除副本.....	147
5.1.9 包周期实例变更规格.....	151
5.2 实例管理.....	158
5.2.1 更新实例带宽弹性伸缩策略.....	158
5.2.2 查询实例带宽弹性伸缩策略.....	166
5.2.3 删除实例带宽弹性伸缩策略.....	173
5.2.4 查询实例参数修改记录详情.....	179
5.2.5 主备切换.....	184
5.2.6 修改密码.....	190
5.2.7 重置密码.....	197
5.2.8 变更可用区.....	205
5.2.9 查询实例状态.....	211

5.2.10 重启实例或清空数据.....	220
5.2.11 查询运行中实例的统计信息.....	226
5.2.12 开启或关闭客户端 IP 透传.....	233
5.2.13 变更指定实例的带宽.....	238
5.2.14 开启/修改实例公网访问.....	242
5.2.15 关闭实例公网访问.....	252
5.2.16 集群分片倒换.....	258
5.2.17 异步交换实例主备节点.....	266
5.2.18 查询实例是否可以扩容.....	272
5.2.19 查询集群实例拓扑关系图.....	280
5.2.20 获取实例分片带宽.....	294
5.2.21 修改实例分片带宽.....	302
5.2.22 异步导出实例资源.....	309
5.2.23 指定实例节点启停开关.....	317
5.2.24 根据实例 ID 获取实例内核版本信息.....	325
5.2.25 升级实例小版本/代理版本.....	333
5.2.26 查询实例重命名的命令.....	339
5.2.27 切换指定实例的子网信息.....	344
5.3 分片与副本.....	350
5.3.1 设置备节点优先级.....	350
5.3.2 域名摘除 IP.....	357
5.3.3 查询分片信息.....	361
5.3.4 批量查询实例节点信息.....	370
5.3.5 查询实例节点信息.....	379
5.3.6 获取副本状态.....	386
5.4 参数管理.....	392
5.4.1 查询实例配置参数.....	392
5.4.2 修改实例配置参数.....	400
5.4.3 异步修改实例配置参数.....	406
5.4.4 查询实例参数修改记录列表.....	411
5.5 备份与恢复.....	418
5.5.1 删除备份文件.....	418
5.5.2 恢复指定实例.....	422
5.5.3 查询实例恢复记录.....	427
5.5.4 获取备份文件下载链接.....	436
5.5.5 备份指定实例.....	443
5.5.6 查询实例备份信息.....	449
5.6 数据迁移.....	458
5.6.1 进行 IP 交换.....	458
5.6.2 IP 交换回滚.....	465
5.6.3 创建备份导入任务.....	472
5.6.4 查询迁移任务列表.....	482

5.6.5 删除数据迁移任务.....	493
5.6.6 查询迁移任务详情.....	500
5.6.7 设置迁移任务自动重连.....	514
5.6.8 停止数据迁移任务.....	521
5.6.9 查询在线迁移进度明细.....	535
5.6.10 创建在线数据迁移任务.....	541
5.6.11 配置在线数据迁移任务.....	549
5.6.12 批量停止数据迁移任务.....	558
5.6.13 同步停止数据迁移任务.....	565
5.6.14 查询迁移日志列表.....	571
5.6.15 批量重启在线迁移任务.....	580
5.7 标签管理.....	587
5.7.1 查询租户所有标签.....	587
5.7.2 批量添加或删除标签.....	592
5.7.3 查询单个实例标签.....	604
5.8 缓存分析.....	611
5.8.1 查询过期 Key 扫描记录.....	611
5.8.2 立刻扫描过期 Key.....	618
5.8.3 创建大 key 分析任务.....	624
5.8.4 查询大 key 分析任务列表.....	633
5.8.5 查询大 key 分析详情.....	642
5.8.6 删除大 key 分析记录.....	652
5.8.7 设置大 key 自动分析配置.....	661
5.8.8 查询大 key 自动分析配置.....	670
5.8.9 创建热 key 分析任务.....	677
5.8.10 查询热 key 分析任务列表.....	686
5.8.11 查询热 key 分析详情.....	696
5.8.12 删除热 key 分析任务.....	706
5.8.13 设置热 key 自动分析配置.....	716
5.8.14 查询热 key 自动分析配置.....	724
5.8.15 创建过期 key 扫描任务.....	731
5.8.16 查询自动扫描配置.....	739
5.8.17 修改自动扫描配置.....	746
5.8.18 下载热 key.....	756
5.9 日志管理.....	761
5.9.1 查询慢日志.....	761
5.9.2 查询 Redis 运行日志列表.....	771
5.9.3 采集 Redis 运行日志.....	781
5.9.4 获取日志下载链接.....	787
5.10 账号管理.....	793
5.10.1 创建 ACL 账号.....	793
5.10.2 查询 ACL 账户列表.....	804

5.10.3 修改 ACL 账号密码.....	813
5.10.4 重置 ACL 账号密码.....	820
5.10.5 修改 ACL 角色.....	828
5.10.6 删除 ACL 账号.....	836
5.10.7 ACL 账号修改备注.....	842
5.11 网络安全.....	849
5.11.1 开启/关闭 SSL.....	849
5.11.2 查询实例 SSL 信息.....	854
5.11.3 下载实例 SSL 证书.....	861
5.11.4 设置 IP 白名单分组.....	866
5.11.5 查询指定实例的 IP 白名单.....	875
5.11.6 设置 IP 白名单分组异步接口.....	879
5.11.7 添加 ip 白名单分组.....	891
5.12 后台任务管理.....	897
5.12.1 查询后台任务列表.....	897
5.12.2 删除后台任务.....	905
5.12.3 查询后台任务详细信息.....	910
5.12.4 查询租户 Job 执行结果.....	917
5.12.5 查询任务中心任务列表.....	923
5.12.6 删除任务中心任务.....	930
5.12.7 提交前置检查任务.....	935
5.12.8 查询实例列表导出任务详情.....	940
5.12.9 等待可维护时间窗取消规格变更.....	945
5.13 实例诊断.....	949
5.13.1 创建实例诊断任务.....	949
5.13.2 查询实例诊断任务列表.....	954
5.13.3 删除诊断记录.....	959
5.13.4 查询指定诊断报告.....	964
5.14 模板管理.....	972
5.14.1 查询参数模板列表.....	972
5.14.2 创建自定义模板.....	983
5.14.3 查询参数模板详情.....	991
5.14.4 修改自定义模板.....	999
5.14.5 删除自定义模板.....	1006
5.15 其他接口.....	1012
5.15.1 查询产品规格.....	1012
5.15.2 查询维护时间窗时间段.....	1028
5.15.3 查询租户配额.....	1032
5.15.4 查询单个主维度下子维度监控对象列表.....	1038
5.15.5 查询主维度信息列表.....	1045
5.15.6 查询可用区信息.....	1050
5.15.7 执行 web-cli 命令.....	1052

5.15.8 登出 WebCli.....	1057
5.15.9 登录 WebCli.....	1063
5.16 会话管理.....	1070
5.16.1 下发查询会话列表任务.....	1070
5.16.2 获取会话列表.....	1075
5.16.3 kill 指定的会话.....	1082
5.16.4 下发 kill 指定节点或实例的全部会话任务.....	1087
5.17 离线全量 key 分析.....	1093
5.17.1 创建离线全量 key 分析任务.....	1093
5.17.2 查询离线全量 key 分析任务列表.....	1102
5.17.3 查询离线全量 key 分析详情.....	1111
5.17.4 删除离线全量 key 分析记录.....	1123
6 权限策略和授权项.....	1131
7 附录.....	1137
7.1 状态码.....	1137
7.2 错误码.....	1139
7.3 获取项目 ID.....	1168
7.4 获取账户名和账号 ID.....	1169
7.5 缓存实例状态说明.....	1170
7.6 实例诊断 ID 说明.....	1170
8 历史 API.....	1172
8.1 生命周期管理.....	1172
8.1.1 创建缓存实例.....	1172
8.1.2 删除实例.....	1183
8.1.3 批量删除实例.....	1184
8.1.4 查询指定实例.....	1186
8.1.5 查询所有实例列表.....	1191
8.1.6 修改实例信息.....	1196
8.1.7 扩容缓存实例.....	1198
8.2 实例管理.....	1200
8.2.1 重启实例或清空数据.....	1200
8.2.2 查询分片信息.....	1202
8.2.3 添加副本.....	1205
8.2.4 查询运行中实例的统计信息.....	1206
8.2.5 查询实例状态.....	1208
8.2.6 修改密码.....	1209
8.3 参数管理.....	1211
8.3.1 修改实例配置参数.....	1212
8.3.2 查询实例配置参数.....	1213
8.4 备份和恢复.....	1220
8.4.1 备份指定实例.....	1220

8.4.2 恢复指定实例.....	1222
8.4.3 查询实例备份信息.....	1223
8.4.4 查询实例恢复记录.....	1226
8.4.5 删除备份文件.....	1229
8.4.6 获取备份文件下载链接.....	1230
8.5 标签管理.....	1232
8.5.1 查询租户所有标签.....	1232
8.5.2 批量添加或删除标签.....	1233
8.5.3 查询单个实例标签.....	1235
8.6 其他接口.....	1236
8.6.1 查询产品规格列表.....	1236
8.6.2 查询租户的试用权限.....	1241
8.6.3 查询租户配额.....	1243
8.6.4 查询维护时间窗时间段.....	1245
8.6.5 查询可用区信息.....	1247
8.7 实例详情.....	1248
8.7.1 变更指定实例的带宽.....	1249

1 使用前必读

欢迎使用分布式缓存服务（Distributed Cache Service）。分布式缓存服务是一款内存数据库服务，兼容了Redis和Memcached两种内存数据库引擎，为您提供即开即用、安全可靠、弹性扩容、便捷管理的在线分布式缓存能力，满足用户高并发及数据快速访问的业务诉求。

您可以使用本文档提供API对分布式缓存服务进行相关操作，如创建、删除、扩容等。支持的全部操作请参见[API概览](#)。

分布式缓存服务提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

须知

- 部分接口仅在指定region支持，请以实际为准。
- DCS服务持续增加新的功能，将不可避免对现有接口有所调整，比如增加响应参数。
- 为了减少接口变更带来的影响，除了DCS服务自身尽量做到接口向下兼容的同时，用户在使用过程中，应当接受出现返回内容（JSON格式）含有未使用的参数和值的现象，即能够正常忽略未使用的参数和值。
- 关于客户端连接缓存实例，请参考用户指南中的[连接Redis缓存实例](#)、[连接Memcached缓存实例](#)，有Redis、Memcached的多语言客户端连接示例说明。

约束与限制

- 您能创建实例数和总内存大小，具体请参见[服务配额](#)。
- 更详细的限制请参见具体API的说明。

终端节点

终端节点即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询服务的终端节点。

基本概念

- 账号

用户注册账号时，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常管理工作。

- 用户

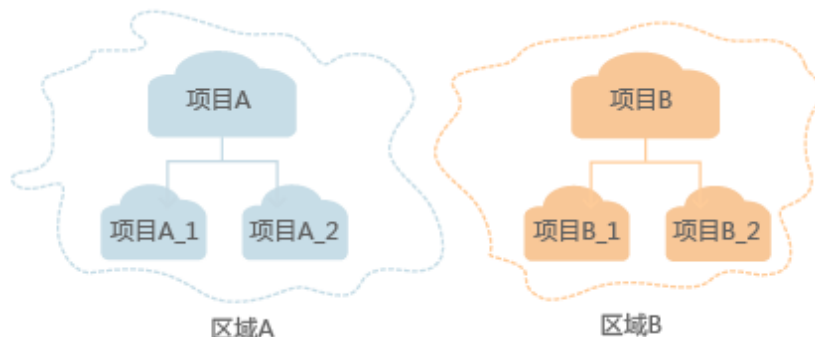
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）：从地理位置和网络时延维度划分，同一个Region内共享弹性计算、对象存储、VPC网络、弹性公网IP等公共服务。同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户或不同地区的要求。
- 可用区：一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 项目

区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中购买资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



同样在[我的凭证](#)下，您可以查看项目ID。

- 企业项目

企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理服务用户指南](#)》。

2 API 概览

表 2-1 分布式缓存服务接口列表

类型	说明
生命周期管理	包括实例的创建、查询和删除，实例信息的修改以及实例扩容。
实例管理	包括重启实例、查询实例状态、修改密码、查询实例的统计信息等接口。
分片与副本	包括添加副本、设置备节点切换为主节点的优先级、查询分片信息等接口。
参数管理	包括查询和修改实例配置参数接口。
备份与恢复	包括备份实例、恢复实例、查看备份信息和恢复记录等接口
数据迁移	包括创建实例迁移任务接口。
标签管理	用户管理实例标签的接口，包括为实例添加标签、删除标签、查询标签以及查询租户所有标签。
缓存分析	包括设置大Key和热Key分析任务、查询大Key和热Key等接口。
日志管理	包括查询慢日志的接口。
ACL账号管理	包括创建、查询、修改、或删除ACL账号的接口。
IP白名单管理	包括为指定实例设置IP白名单分组、查询实例的IP白名单的接口。
后台任务管理	包括查询后台任务列表、删除后台任务的接口。
实例诊断	包括查询可用区信息、产品规格列表以及维护时间窗等接口。
模板管理	包含查询参数模板列表、创建自定义模板接口。
其他接口	包括查询产品规格、查询维护时间窗时间段、查询可用区信息等。

3 如何调用 API

3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的[管理员创建IAM用户](#)接口说明如何调用API。

您还可以通过这个视频教程了解如何构造请求调用API：<https://bbs.huaweicloud.com/videos/102987>。

请求 URI

请求URI由如下部分组成。

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **URI-scheme**：表示用于传输请求的协议，当前所有API均采用HTTPS协议。
- **Endpoint**：指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从[地区和终端节点](#)获取。例如IAM服务在“华北-北京四”区域的Endpoint为“iam.cn-north-4.myhuaweicloud.com”。
- **resource-path**：资源路径，也即API访问路径。从具体API的URI模块获取，例如“管理员创建IAM用户”接口的resource-path为“/v3.0/OS-USER/users”。
- **query-string**：查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“?limit=10”，表示查询不超过10条数据。

例如您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint（比如“华北-北京四”区域的Endpoint：“iam.cn-north-4.myhuaweicloud.com”），并在[管理员创建IAM用户](#)的URI部分找到resource-path（/v3.0/OS-USER/users），拼接起来如下所示。

```
https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
```

图 3-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **GET:** 请求服务器返回指定资源。
- **PUT:** 请求服务器更新指定资源。
- **POST:** 请求服务器新增资源或执行特殊操作。
- **DELETE:** 请求服务器删除指定资源，如删除对象等。
- **HEAD:** 请求服务器资源头部。
- **PATCH:** 请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在[管理员创建IAM用户](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

- **Content-Type:** 消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **Authorization:** 签名认证信息，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。AK/SK认证的详细说明请参考[AK/SK认证](#)。
- **X-Sdk-Date:** 请求发送的时间，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。AK/SK认证的详细说明请参考[AK/SK认证](#)。
- **X-Auth-Token:** 用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token也就是调用[获取用户Token](#)接口的响应值，该接口是唯一不需要认证的接口。
- **X-Project-ID:** 子项目ID，可选，在多项目场景中使用。如果云服务资源创建在子项目中，AK/SK认证方式下，操作该资源的接口调用需要在请求消息头中携带X-Project-ID。

- **X-Domain-ID**: 账号ID, 可选。AK/SK认证方式下, 全局服务的接口调用时, 需在请求消息头中携带X-Domain-ID。

对于**管理员创建IAM用户**接口, 使用AK/SK方式认证时, 添加消息头后的请求如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

请求消息体

请求消息体通常以结构化格式发出, 与请求消息头中Content-type对应, 传递除请求消息头之外的内容。若请求消息体中参数支持中文, 则中文字符必须为UTF-8编码, 并在Content-type中声明字符编码方式, 例如: Content-Type: application/json; charset=utf-8。

每个接口的请求消息体内容不同, 也并不是每个接口都需要有请求消息体 (或者说消息体为空), **GET**、**DELETE**操作类型的接口就不需要消息体, 消息体具体内容需要根据具体接口而定。

对于**管理员创建IAM用户**接口, 您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示, 其中加粗的斜体字段需要根据实际值填写。

- **accountid**为IAM用户所属的账号ID。
- **username**为要创建的IAM用户名。
- **email**为IAM用户的邮箱。
- *********为IAM用户的登录密码。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****

{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

到此为止, 一个API请求所需要的内容已经准备完成, 您可以使用curl、Postman或直接编写代码等方式发送请求调用API。

3.2 认证鉴权

调用接口有如下两种认证方式, 您可以选择其中一种进行认证鉴权。

- **AK/SK认证**: 通过AK (Access Key ID) /SK (Secret Access Key) 加密调用请求。推荐使用AK/SK认证, 其安全性比Token认证要高。
- **Token认证**: 通过Token认证调用请求。

AK/SK 认证

说明

- AK/SK 签名认证方式仅支持消息体大小 12MB 以内，12MB 以上的请求请使用 Token 认证。
- AK/SK 既可以使用永久访问密钥中的 AK/SK，也可以使用临时访问密钥中的 AK/SK，但使用临时访问密钥的 AK/SK 时需要额外携带 “X-Security-Token” 字段，字段值为临时访问密钥的 security_token。

AK/SK 认证就是使用 AK/SK 对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK (Access Key ID)：访问密钥 ID。与私有访问密钥关联的唯一标识符；访问密钥 ID 和私有访问密钥一起使用，对请求进行加密签名。
- SK (Secret Access Key)：与访问密钥 ID 结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用 AK/SK 认证时，您可以基于签名算法使用 AK/SK 对请求进行签名，也可以使用专门的签名 SDK 对请求进行签名。详细的签名方法和 SDK 使用方法请参见 [API 签名指南](#)。

须知

签名 SDK 只提供签名功能，与服务提供的 SDK 不同，使用时请注意。

您也可以通过这个视频教程了解 AK/SK 认证的使用：<https://bbs.huaweicloud.com/videos/100697>。

Token 认证

说明

- Token 的有效期为 24 小时，需要使用一个 Token 鉴权时，可以先缓存起来，避免频繁调用。
- 使用 Token 前请确保 Token 离过期有足够的时间，防止调用 API 的过程中 Token 过期导致调用 API 失败。

Token 在计算机系统中代表令牌（临时）的意思，拥有 Token 就代表拥有某种权限。Token 认证就是在调用 API 的时候将 Token 加到请求消息头，从而通过身份认证，获得操作 API 的权限。

Token 可通过调用 [获取用户 Token](#) 接口获取，调用本服务 API 需要 project 级别的 Token，即调用 [获取用户 Token](#) 接口时，请求 body 中 auth.scope 的取值需要选择 project，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    }
  }
}
```



```
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFG....”加到请求消息头即可，如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFG....
```

您还可以通过这个视频教程了解如何使用Token认证：<https://bbs.huaweicloud.com/videos/101333>。

3.3 返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于[管理员创建IAM用户](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[管理员创建IAM用户](#)接口，返回如[图3-2](#)所示的消息头。

图 3-2 管理员创建 IAM 用户响应消息头

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_ _null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于[管理员创建IAM用户](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "user": {
    "id": "c131886aec...",
    "name": "IAMUser",
    "description": "IAM User Description",
    "areacode": "",
    "phone": "",
    "email": "****@***.com",
    "status": null,
    "enabled": true,
    "pwd_status": false,
    "access_mode": "default",
    "is_domain_owner": false,
    "xuser_id": "",
    "xuser_type": "",
    "password_expires_at": null,
    "create_time": "2024-05-21T09:03:41.000000",
    "domain_id": "d78cbac1.....",
    "xdomain_id": "30086000.....",
    "xdomain_type": "",
    "default_project_id": null
  }
}
```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_msg": "Request body is invalid.",
  "error_code": "IAM.0011"
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4应用示例

4.1 创建 DCS 高频场景示例

本节介绍了使用API购买DCS实例过程中的一些常见问题及处理方法。

表 4-1 购买 DCS 实例常见问题及处理方法

常见问题	处理方法
购买包周期实例	为您介绍购买包周期实例时参数设置。
删除包周期实例	包周期资源需要通过退订的方式才可以被删除。
使用API购买实例时设置副本数	使用API购买的实例，副本数的值取决于在购买时输入的spec_code参数，为调用 查询产品规格 响应示例中replica_count的值。
获取子网ID	通过查询子网ID接口，获取正确的ID供购买实例时配置。
购买实例时可用区资源不足	可用区资源不足会导致创建实例失败，可通过调用 查询产品规格 响应示例中flavors_available_zones.az_codes的值来确保在购买实例时选择充足资源的可用区。
查看资源的可用配额	通过调用 查询租户配额 接口，通过quota值可确定可以创建的实例最大数和总内存的配额限制。

购买包周期实例

您可使用[创建缓存实例](#)接口购买包周期实例，其与购买按需实例的区别在于，购买实例的body体中，将bss_param.charging_mode字段配置为“prePaid”，

bss_param的详细介绍请参考[表3 BssParam](#)。

如下请求体所示，在cn-north-4区域购买一个包周期DCS实例，时长为一个月，且下单后不自动续订，不自动付费。

```
{
  "name": "test",
  "engine": "Redis",
  "engine_version": "5.0",
  "capacity": 2,
  "spec_code": "redis.ha.xu1.large.r2.2",
  "az_codes": [
    "cn-north-4a"
  ],
  "vpc_id": "c29eade3-05d3-41a3-84e4-2cc1f3057502",
  "subnet_id": "b573d369-2b49-4595-8727-2408244a7675",
  "no_password_access": true,
  "bss_param": {
    "charging_mode": "prePaid",
    "is_auto_pay": "false",
    "period_type": "month",
    "period_num": 2
  },
  "instance_backup_policy": {
    "backup_type": "auto",
    "save_days": 2,
    "periodical_backup_plan": {
      "backup_at": [
        2
      ],
      "period_type": "weekly",
      "begin_at": "00:00-01:00"
    }
  },
  "tags": [
    {
      "key": "test",
      "value": "123"
    }
  ]
}
```

购买任务提交成功后，会返回以下order_id，即订单ID。

```
{
  "instance_id": "23be3a34-8cc1-4eab-9ad2-b085eadee9c4",
  "order_id": "CS2111091732PY0DE",
  "instances": [
    {
      "instance_id": "23be3a34-8cc1-4eab-9ad2-b085eadee9c4",
      "instance_name": "test"
    }
  ]
}
```

上面请求体中bss_param.is_auto_pay取值为false或不填该字段则需要手动去支付，手动支付可以填写优惠券和折扣券等信息。手动支付需要调用[支付包年/包月产品订单](#)支付，示例如下。

```
POST https://bss.myhuaweicloud.com/v2/orders/customer-orders/pay
{
  "order_id": "CS2111091732PY0DE"
}
```

删除包周期实例

包周期的DCS实例无法直接删除，需要调用[退订包年/包月资源](#)接口进行退订。

```
POST https://bss.myhuaweicloud.com/v2/orders/subscriptions/resources/unsubscribe
{
  "resource_ids": [
    "44b393c8-e4e4-4dcb-bad3-54858dcc72c7"
  ],
  "unsubscribe_type": 1
}
```

其中resource_ids表示资源ID，对退订DCS来说，就是购买包周期DCS时返回的instance_id。

使用 API 购买实例时设置副本数

使用API购买实例后，副本数的值取决于在购买时输入的spec_code参数。副本数即为[查询产品规格](#)接口响应示例的replica_count值。

请求示例：

```
GET https://{dcs_endpoint}/v2/flavors?
cache_mode={cache_mode}&engine={engine}&engine_version={engine_version}&cpu_type={cpu_type}&capacity={capacity}
```

响应示例：

```
{
  "flavors": [
    {
      "dec": false,
      "spec_code": "redis.ha.xu1.large.r4.2",
      "cloud_service_type_code": "hws.service.type.dcs",
      "cloud_resource_type_code": "hws.resource.type.dcs3",
      "cache_mode": "ha",
      "engine": "redis",
      "engine_version": "4.0;5.0",
      "product_type": "generic",
      "cpu_type": "x86_64",
      "storage_type": "DRAM",
      "capacity": [
        "2"
      ],
      "billing_mode": [
        "Hourly",
        "Monthly",
        "Yearly"
      ],
      "tenant_ip_count": 4,
      "pricing_type": "normal",
      "is_dec": false,
      "attrs": [
        {
          "capacity": "2",
          "name": "max_memory",
          "value": "2"
        },
        {
          "capacity": "2",
          "name": "max_connections",
          "value": "50000"
        },
        {
          "capacity": "2",
          "name": "sharding_num",
          "value": "1"
        },
        {
          "capacity": "2",
          "name": "proxy_num",
          "value": "0"
        }
      ]
    }
  ]
}
```

```
},
{
  "capacity": "2",
  "name": "db_number",
  "value": "256"
},
{
  "capacity": "2",
  "name": "max_clients",
  "value": "10000"
},
{
  "capacity": "2",
  "name": "max_bandwidth",
  "value": "128"
}
],
"flavors_available_zones": [
{
  "capacity": "2",
  "unit": "GB",
  "available_zones": [
    "a0865121f83b41cbafce65930a22a6e8",
    "15dcd3f789cb4fc088dbdece9c5eb547",
    "2dcb154ac2724a6d92e9bcc859657c1e",
    "effdcbc7d4d64a02aa1fa26b42f56533"
  ],
  "az_codes": [
    "xx-xx-xx1",
    "xx-xx-xx2",
    "xx-xx-xx3",
    "xx-xx-xx4"
  ]
}
],
"replica_count": 4,
"inquiry_spec_code": null
}
]
```

说明

在购买实例时，输入的spec_code参数须与capacity参数一一对应。

获取子网 ID

在购买实例时配置正确的子网ID，才能够成功连接实例。可通过GET /v1/{project_id}/subnets?{vpc_id}获取准确子网ID，其中vpc_id可通过GET /v1/{project_id}/vpcs接口获取。

获取vpc_id的请求示例：

```
GET https://{vpc_endpoint}/v1/7d80ae32f57b499eb8781f9a9f57c538/vpcs
```

{vpc_endpoint}信息，请从[地区和终端节点](#)获取。

获取vpc_id的响应示例：

```
{
  "vpcs": [{
    "id": "743bf021-2c2d-4511-aeac-85bd48c06af7",
    "name": "vpc-d2d4",
    "description": "",
    "cidr": "192.168.0.0/16",
    "status": "OK",
    "routes": [],
    "enterprise_project_id": "0"
  ]
}
```

```
    }}  
  }
```

获取子网ID的请求示例：

```
GET https://{vpc_endpoint}/v1/7d80ae32f57b499eb8781f9a9f57c538/subnets?vpc_id=743bf021-2c2d-4511-aeac-85bd48c06af7
```

获取子网ID的响应示例：

```
{  
  "subnets": [{  
    "id": "b0d6e0ac-fcce-4e11-a4a6-707e851ae1c3",  
    "name": "subnet-d2e4",  
    "description": "",  
    "cidr": "192.168.0.0/24",  
    "dnsList": ["100.125.1.250",  
    "100.125.129.250"],  
    "status": "ACTIVE",  
    "vpc_id": "743bf021-2c2d-4511-aeac-85bd48c06af7",  
    "ipv6_enable": false,  
    "gateway_ip": "192.168.0.1",  
    "dhcp_enable": true,  
    "primary_dns": "100.125.1.250",  
    "secondary_dns": "100.125.129.250",  
    "availability_zone": "xxxxxx",  
    "neutron_network_id": "b0d6e0ac-fcce-4e11-a4a6-707e851ae1c3",  
    "neutron_subnet_id": "3d4ccce0-cadc-4af4-8c21-14d2857ffe5e",  
    "extra_dhcp_opts": []  
  }]  
}
```

购买实例时可用区资源不足

在购买实例时，如需要查询某一规格在某可用区是否资源充足，可通过调用[查询产品规格](#)查看规格所在可用区的详细信息，通过flavors_available_zones的值判断该规格可选的可用区。

如在某region、规格为redis.ha.xu1.large.r2.2的资源充足可用区的信息，请求示例：

```
https://{dcs_endpoint}/v2/05041fffa40025702f6dc009cc6f8f33/flavors?spec_code=redis.ha.xu1.large.r2.2
```

响应示例：

```
{  
  "flavors": [  
    {  
      "dec": false,  
      "spec_code": "redis.ha.xu1.large.r2.2",  
      "cloud_service_type_code": "hws.service.type.dcs",  
      "cloud_resource_type_code": "hws.resource.type.dcs3",  
      "cache_mode": "ha",  
      "engine": "redis",  
      "engine_version": "4.0;5.0",  
      "product_type": "generic",  
      "cpu_type": "x86_64",  
      "storage_type": "DRAM",  
      "capacity": [  
        "2"  
      ],  
      "billing_mode": [  
        "Hourly",  
        "Monthly",  
        "Yearly"  
      ],  
      "tenant_ip_count": 2,  
      "pricing_type": "normal",  
      "is_dec": false,  
    }  
  ]  
}
```

```
"attrs": [
  {
    "capacity": "2",
    "name": "max_memory",
    "value": "2"
  },
  {
    "capacity": "2",
    "name": "max_connections",
    "value": "50000"
  },
  {
    "capacity": "2",
    "name": "sharding_num",
    "value": "1"
  },
  {
    "capacity": "2",
    "name": "proxy_num",
    "value": "0"
  },
  {
    "capacity": "2",
    "name": "db_number",
    "value": "256"
  },
  {
    "capacity": "2",
    "name": "max_clients",
    "value": "10000"
  },
  {
    "capacity": "2",
    "name": "max_bandwidth",
    "value": "128"
  }
],
"flavors_available_zones": [
  {
    "capacity": "2",
    "unit": "GB",
    "available_zones": [
      "a0865121f83b41cbafce65930a22a6e8",
      "15dcd3f789cb4fc088dbdece9c5eb547",
      "2dc154ac2724a6d92e9bcc859657c1e",
      "effdcbc7d4d64a02aa1fa26b42f56533"
    ],
    "az_codes": [
      "xx-xx-xx1",
      "xx-xx-xx2",
      "xx-xx-xx3",
      "xx-xx-xx4"
    ]
  }
],
"replica_count": 2,
"inquiry_spec_code": null
}
]
```

其中flavors_available_zones罗列出了该规格资源充足的可用区。

查看资源的可用配额

如需查询当前账号的资源配额信息，包括已使用的配额，可以通过调用[查询租户配额](#)接口。

请求示例：

```
GET https://{dcs_endpoint}/v2/05041ffa40025702f6dc009cc6f8f33/quota
```

响应示例：

```
{
  "quotas": {
    "resources": [
      {
        "quota": 160,
        "used": 1,
        "type": "instance",
        "min": 0,
        "max": 2147483647,
        "unit": null
      },
      {
        "quota": 64000,
        "used": 0,
        "type": "ram",
        "min": 0,
        "max": 1048576,
        "unit": "GB"
      }
    ]
  }
}
```

- 当type为instance时，max表示可申请实例配额的最大值，min表示可申请实例配额的最小值。
- 当type为ram时，max表示可申请内存配额的最大值，min表示可申请内存配额的最小值。
- quota为可以创建的实例最大数和总内存的配额限制。
- used为已创建的实例个数和已使用的内存配额。

4.2 示例 1：创建 Redis 实例

场景描述

本章节指导用户通过API创建Redis实例。API的调用方法请参见[如何调用API](#)。

本文以创建一个实例规格为2G、缓存版本为Redis5.0、实例类型为单机、CPU架构为ARM、免密访问方式的缓存实例。

涉及接口

创建Redis实例时，需要进行查询产品规格、查询VPC、子网等操作，涉及的API如下：

- [查询VPC ID](#)：确定待创建DCS实例的VPC ID。
- [查询子网ID](#)：确定待创建DCS实例的子网ID。
- [查询产品规格](#)：确定待创建DCS实例的产品规格。
- [创建实例](#)：创建指定规格的DCS实例。
- [查询创建结果](#)：确认创建DCS实例是否成功。

步骤 1：确定待创建实例的 VPC

步骤1 查询VPC列表。

- 接口相关信息
URI格式：GET /v1/{project_id}/vpcs
- 调试
您可以在[API Explorer](#)中调试该接口。
- 请求示例
GET https://{vpc_endpoint}/v1/7d80ae32f57b499eb8781f9a9f57c538/vpcs
{vpc_endpoint}信息，请从[地区和终端节点](#)获取。
- 响应示例

```
{
  "vpcs": [{
    "id": "743bf021-2c2d-4511-aeac-85bd48c06af7",
    "name": "vpc-d2d4",
    "description": "",
    "cidr": "192.168.0.0/16",
    "status": "OK",
    "routes": [],
    "enterprise_project_id": "0"
  }]
}
```

步骤2 根据需要，记录VPC ID。

----结束

步骤 2：确定待创建实例的子网

步骤1 查询子网列表。

- 接口相关信息
URI格式：GET /v1/{project_id}/subnets?vpc_id
- 调试
您可以在[API Explorer](#)中调试该接口。
- 请求示例
GET https://{vpc_endpoint}/v1/7d80ae32f57b499eb8781f9a9f57c538/subnets?vpc_id=743bf021-2c2d-4511-aeac-85bd48c06af7
{vpc_endpoint}信息，请从[地区和终端节点](#)获取。
- 响应示例

```
{
  "subnets": [{
    "id": "b0d6e0ac-fcce-4e11-a4a6-707e851ae1c3",
    "name": "subnet-d2e4",
    "description": "",
    "cidr": "192.168.0.0/24",
    "dnsList": ["100.125.1.250",
    "100.125.129.250"],
    "status": "ACTIVE",
    "vpc_id": "743bf021-2c2d-4511-aeac-85bd48c06af7",
    "ipv6_enable": false,
    "gateway_ip": "192.168.0.1",
    "dhcp_enable": true,
    "primary_dns": "100.125.1.250",
    "secondary_dns": "100.125.129.250",
    "availability_zone": "cn-southwest-2a",
  }]
}
```

```

    "neutron_network_id": "b0d6e0ac-fcce-4e11-a4a6-707e851ae1c3",
    "neutron_subnet_id": "3d4ccce0-cadc-4af4-8c21-14d2857ffe5e",
    "extra_dhcp_opts": []
  }
}

```

步骤2 根据需要，记录子网ID。

----结束

步骤 3：确定待创建实例的产品规格

步骤1 查询产品规格。

- 接口相关信息

URI格式：GET /v2/{project_id}/flavors?

spec_code={spec_code}&cache_mode={cache_mode}&engine={engine}&engine_version={engine_version}&cpu_type={cpu_type}&capacity={capacity}

- 调试

您可以在[API Explorer](#)中调试该接口。

- 请求示例

GET https://{dcs_endpoint}/v2/666486c2d9b948c1bba57e714d744fa/flavors?
cache_mode=single&engine=Redis&engine_version=5.0&cpu_type=aarch64&capacity=2

{dcs_endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```

[[
  {
    "dec": false,
    "spec_code": "redis.single.au1.large.2",
    "cloud_service_type_code": "hws.service.type.dcs",
    "cloud_resource_type_code": "hws.resource.type.dcs3",
    "cache_mode": "single",
    "engine": "redis",
    "engine_version": "4.0;5.0",
    "product_type": "generic",
    "cpu_type": "aarch64",
    "storage_type": "DRAM",
    "capacity": ["2"],
    "billing_mode": ["Hourly",
      "RI",
      "Monthly",
      "Yearly"],
    "tenant_ip_count": 1,
    "pricing_type": "normal",
    "is_dec": false,
    "attrs": [{
      "capacity": "2",
      "name": "max_memory",
      "value": "2"
    },
    {
      "capacity": "2",
      "name": "max_connections",
      "value": "10000"
    },
    {
      "capacity": "2",
      "name": "sharding_num",
      "value": "1"
    },
    {
      "capacity": "2",

```

```

    "name": "proxy_num",
    "value": "0"
  },
  {
    "capacity": "2",
    "name": "db_number",
    "value": "256"
  },
  {
    "capacity": "2",
    "name": "max_clients",
    "value": "10000"
  },
  {
    "capacity": "2",
    "name": "max_bandwidth",
    "value": "128"
  }
},
"flavors_available_zones": [{
  "capacity": "2",
  "unit": "GB",
  "available_zones": ["a0865121f83b41cbafce65930a22a6e8",
    "effdcbc7d4d64a02aa1fa26b42f56533"],
  "az_codes": ["cn-north-4b",
    "cn-north-4a"]
}]
}]

```

步骤2 根据需要选择产品类型，并记录产品规格编码和有资源的可用区编码。

----结束

步骤 4：创建实例

- 接口相关信息

URI格式：POST /v2/{project_id}/instances

- 调试

您可以在[API Explorer](#)中调试该接口。

- 请求示例

POST https://{dcs_endpoint}/v2/666486c2d9b948c1bba57e714d744fa/instances

{dcs_endpoint}信息请从[地区和终端节点](#)获取。

Body:

```

{
  "az_codes": ["cn-north-4a"],
  "capacity": 2,
  "engine": "Redis",
  "engine_version": "5.0",
  "name": "dcs-api-test",
  "no_password_access": true,
  "security_group_id": "1982d3a8-67a0-4fc9-a850-bc42a26ba2c0",
  "spec_code": "redis.single.xu1.large.2",
  "subnet_id": "b0d6e0ac-fcce-4e11-a4a6-707e851ae1c3",
  "vpc_id": "743bf021-2c2d-4511-aeac-85bd48c06af7"
}

```

- 响应示例

```

{
  "instances": [{
    "instance_id": "21bc7b53-2494-4f10-bb0b-c0b913d9e329",
    "instance_name": "dcs-api-test"
  }]
}

```

创建Redis实例的参数说明请参考[创建缓存实例](#)。

步骤 5：查询创建结果

- 接口相关信息

URI格式：GET /v2/{project_id}/instances/{instance_id}

- 调试

您可以在[API Explorer](#)中调试该接口。

- 请求示例

GET https://{dcs_endpoint}/v2/666486c2d9b948c1bbea57e714d744fa/instances/21bc7b53-2494-4f10-bb0b-c0b913d9e329

{dcs_endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{
  "free": null,
  "max_memory": 2048,
  "used_memory": 1,
  "instance_id": "21bc7b53-2494-4f10-bb0b-c0b913d9e329",
  "name": "dcs-api-test",
  "resource_spec_code": "redis.single.au1.large.2",
  "engine": "Redis",
  "engine_version": "5.0",
  "internal_version": null,
  "charging_mode": 0,
  "capacity": 2,
  "capacity_minor": null,
  "vpc_id": "743bf021-2c2d-4511-aeac-85bd48c06af7",
  "vpc_name": "vpc-d2d4",
  "ip": "192.168.0.100",
  "domain_name": "redis-88a7bbb-dcs-api-test.dcs.huaweicloud.com",
  "readonly_domain_name": null,
  "port": 6379,
  "status": "RUNNING",
  "freeze_scene": null,
  "created_at": "2020-06-05T03:30:36.273Z",
  "update_at": "2020-06-05T03:30:47.231Z",
  "error_code": null,
  "user_id": "d53977d1adfb49c5b025ba7d33a13fd7",
  "user_name": "paas_dcs_a00421997_02",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00",
  "no_password_access": "true",
  "access_user": null,
  "enable_publicip": false,
  "publicip_id": null,
  "publicip_address": null,
  "enable_ssl": false,
  "service_upgrade": false,
  "service_task_id": "",
  "enterprise_project_id": "0",
  "security_group_id": null,
  "tags": [],
  "product_type": "generic",
  "cpu_type": "aarch64",
  "storage_type": "DRAM",
  "launched_at": "2020-06-05T03:30:47.238Z",
  "is_free": null,
  "libos": false,
  "cache_mode": "single",
  "available_zones": ["d539378ec1314c85b76fefa3f7071458"],
  "subnet_id": "a4112635-3ec0-471c-95c3-5cf49b9533af",
  "backend_addrs": null,
  "crr_role": null,
```

```
{
  "cloud_service_type_code": "hws.service.type.dcs",
  "cloud_resource_type_code": "hws.resource.type.dcs3",
  "support_slow_log_flag": null,
  "ipv6": null,
  "enable_ipv6": false,
  "description": "",
  "product_id": "redis.single.au1.large.2-h",
  "security_group_name": null,
  "subnet_name": "dcs-beta",
  "order_id": null,
  "subnet_cidr": "192.168.0.0/24",
  "task": null,
  "instance_backup_policy": null,
  "enterprise_project_name": null
}
```

实例状态为RUNNING时，即创建成功。

4.3 示例 2：创建数据迁移任务

场景描述

本章节指导用户通过API创建数据迁移任务。API的调用方法请参见[如何调用API](#)。

本文使用在线迁移、全量迁移的方式为例创建数据迁移任务，源实例与目标实例是华为云上处于相同VPC的Redis。

涉及接口

创建数据迁移任务，涉及的API如下：

- [查询源实例的IP和端口](#)：确定迁移的源Redis实例的IP和端口。
- [创建数据迁移任务](#)：创建数据迁移任务。
- [查询迁移结果](#)：确认迁移是否成功。

步骤 1：确定源实例的 IP 和端口

步骤1 查询指定实例的IP和端口。

- 接口相关信息
URI格式：GET /v2/{project_id}/instances/{instance_id}
- 调试
您可以在[API Explorer](#)中调试该接口。
- 请求示例
GET https://{dcs_endpoint}/v2/666486c2d9b948c1bbea57e714d744fa/instances/21bc7b53-2494-4f10-bb0b-c0b913d9e329
{dcs_endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{
  "free": null,
  "max_memory": 2048,
  "used_memory": 1,
  "instance_id": "21bc7b53-2494-4f10-bb0b-c0b913d9e329",
  "name": "dcs-api-test",
  "resource_spec_code": "redis.single.au1.large.2",
  "engine": "Redis",
}
```

```
{
  "engine_version": "5.0",
  "internal_version": null,
  "charging_mode": 0,
  "capacity": 2,
  "capacity_minor": null,
  "vpc_id": "743bf021-2c2d-4511-aeac-85bd48c06af7",
  "vpc_name": "vpc-d2d4",
  "ip": "192.168.0.100",
  "domain_name": "redis-88a7bbb-dcs-api-test.dcs.huaweicloud.com",
  "readonly_domain_name": null,
  "port": 6379,
  "status": "RUNNING",
  "freeze_scene": null,
  "created_at": "2020-06-05T03:30:36.273Z",
  "update_at": "2020-06-05T03:30:47.231Z",
  "error_code": null,
  "user_id": "d53977d1adfb49c5b025ba7d33a13fd7",
  "user_name": "paas_dcs_a00421997_02",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00",
  "no_password_access": "true",
  "access_user": null,
  "enable_publicip": false,
  "publicip_id": null,
  "publicip_address": null,
  "enable_ssl": false,
  "service_upgrade": false,
  "service_task_id": "",
  "enterprise_project_id": "0",
  "security_group_id": null,
  "tags": [],
  "product_type": "generic",
  "cpu_type": "aarch64",
  "storage_type": "DRAM",
  "launched_at": "2020-06-05T03:30:47.238Z",
  "is_free": null,
  "libos": false,
  "cache_mode": "single",
  "available_zones": ["d539378ec1314c85b76fefa3f7071458"],
  "subnet_id": "a4112635-3ec0-471c-95c3-5cf49b9533af",
  "backend_addrs": null,
  "crr_role": null,
  "cloud_service_type_code": "hws.service.type.dcs",
  "cloud_resource_type_code": "hws.resource.type.dcs3",
  "support_slow_log_flag": null,
  "ipv6": null,
  "enable_ipv6": false,
  "description": "",
  "product_id": "redis.single.au1.large.2-h",
  "security_group_name": null,
  "subnet_name": "dcs-beta",
  "order_id": null,
  "subnet_cidr": "192.168.0.0/24",
  "task": null,
  "instance_backup_policy": null,
  "enterprise_project_name": null
}
```

步骤2 根据需要，记录实例的IP和端口。

----结束

步骤 2：创建数据迁移任务

- 接口相关信息
URI格式：POST /v2/{project_id}/migration-task
- 调试

您可以在[API Explorer](#)中调试该接口。

- 请求示例

POST https://{dcs_endpoint}/v2/666486c2d9b948c1bba57e714d744fa/migration-task

{dcs_endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{
  "migration_method": "full_amount_migration",
  "migration_type": "online_migration",
  "network_type": "vpc",
  "source_instance": {
    "addr": "192.168.0.79:6379"
  },
  "target_instance": {
    "id": "92b2b922-0f96-4664-acc5-04d922867c4d"
  },
  "task_name": "dcs-test-migration"
}
```

- 响应示例

```
{
  "id": "8aa6999f727f0a40017283479d730d7c",
  "name": "dcs-test-migration",
  "status": "MIGRATING",
  "error": ""
}
```

创建数据迁移任务的参数说明请参考[创建备份导入任务](#)。

步骤 3：查询迁移结果

- 接口相关信息

URI格式：GET /v2/{project_id}/migration-task/{task_id}

- 调试

您可以在[API Explorer](#)中调试该接口。

- 请求示例

GET https://{dcs_endpoint}/v2/666486c2d9b948c1bba57e714d744fa/migration-task/8aa6999f727f0a40017283479d730d7c

{dcs_endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{
  "task_name": "dcs-test-migration",
  "task_id": "8aa6999f727f0a40017283479d730d7c",
  "description": null,
  "status": "SUCCESS",
  "migration_type": "online_migration",
  "migration_method": "full_amount_migration",
  "backup_files": null,
  "network_type": "vpc",
  "bandwidth_limit_mb": null,
  "source_instance": {
    "id": null,
    "ip": "192.168.0.79:6379",
    "port": null,
    "name": null,
    "password": null,
    "addr": null
  },
  "target_instance": {
    "id": "92b2b922-0f96-4664-acc5-04d922867c4d",

```

```
{
  "ip": null,
  "port": null,
  "name": "dcs-xwvq",
  "password": null,
  "addrs": null
},
"created_at": "2020-06-05T07:00:08.204Z",
"updated_at": "2020-06-05T07:03:25.075Z"
}
```

任务状态为SUCCESS时，即迁移成功。

4.4 示例 3：批量删除 Redis 实例

场景描述

本章节指导用户通过API批量删除Redis实例。API的调用方法请参见[如何调用API](#)。

涉及接口

批量删除Redis实例时，需要进行查询待删除实例的ID，涉及的API如下：

- [查询实例ID](#)：查询待删除Redis实例的ID。
- [批量删除实例](#)：批量删除Redis实例。

步骤 1：查询待删除实例的 ID

步骤1 查询实例ID。

- 接口相关信息
URI格式：GET /v2/{project_id}/instances
- 调试
您可以在[API Explorer](#)中调试该接口。
- 请求示例
GET https://{dcs_endpoint}/v2/7d80ae32f57b499eb8781f9a9f57c538/instances
{dcs_endpoint}信息，请从[地区和终端节点](#)获取。
- 响应示例

```
{
  "instances": [{
    "free": null,
    "max_memory": 1024,
    "used_memory": 2,
    "instance_id": "aabe73af-1db8-4401-b39c-c56996023989",
    "name": "dcs-8b1c-000",
    "resource_spec_code": "redis.ha.xu1.large.r2.1",
    "engine": "Redis",
    "engine_version": "5.0",
    "internal_version": null,
    "charging_mode": 0,
    "capacity": 1,
    "capacity_minor": null,
    "vpc_id": "cec3dca1-3700-4305-8eb0-5669c01f05b6",
    "vpc_name": null,
    "ip": "192.168.0.12",
    "domainName": "redis-aabe73a-dcs-8b1c-000.dcs.huaweicloud.com",
    "domain_name": "redis-aabe73a-dcs-8b1c-000.dcs.huaweicloud.com",
    "readonlyDomainName": null,
    "readonly_domain_name": null,
  ]
}
```



```

"port": 6379,
"status": "RUNNING",
"freeze_scene": null,
"created_at": "2020-06-18T03:11:30.615Z",
"update_at": "2020-06-18T03:11:58.936Z",
"error_code": null,
"user_id": "05040c67cc00256f1f28c009a1c2a0f7",
"user_name": "dcstest",
"maintain_begin": "02:00:00",
"maintain_end": "06:00:00",
"no_password_access": "true",
"access_user": null,
"enable_publicip": false,
"publicip_id": null,
"publicip_address": null,
"enable_ssl": false,
"service_upgrade": false,
"service_task_id": "",
"enterprise_project_id": "0",
"security_group_id": null,
"tags": null,
"product_type": "generic",
"cpu_type": "x86_64",
"storage_type": "DRAM",
"task_status": null,
"launched_at": "2020-06-18T03:11:58.945Z",
"is_free": null,
"libos": false,
"cache_mode": "ha",
"available_zones": ["dfeb4826256b47aa828e72d7522e108c",
"043c7e39ecb347a08dc8fcb6c35a274e"],
"subnet_id": "e6eb5ff4-da2a-4323-bddc-c7b179f9aa1e",
"backend_addrs": null,
"crr_role": null,
"cloud_service_type_code": "hws.service.type.dcs",
"cloud_resource_type_code": "hws.resource.type.dcs3",
"support_slow_log_flag": null,
"ipv6": null,
"enable_ipv6": false,
"description": ""
},
{
  "free": null,
  "max_memory": 1024,
  "used_memory": 2,
  "instance_id": "9d78d0f4-a110-4bcd-8ca3-5aec414163c1",
  "name": "dcs-8b1c-001",
  "resource_spec_code": "redis.ha.xu1.large.r2.1",
  "engine": "Redis",
  "engine_version": "5.0",
  "internal_version": null,
  "charging_mode": 0,
  "capacity": 1,
  "capacity_minor": null,
  "vpc_id": "cec3dca1-3700-4305-8eb0-5669c01f05b6",
  "vpc_name": null,
  "ip": "192.168.0.33",
  "domainName": "redis-9d78d0f-dcs-8b1c-001.dcs.huaweicloud.com",
  "domain_name": "redis-9d78d0f-dcs-8b1c-001.dcs.huaweicloud.com",
  "readonlyDomainName": null,
  "readonly_domain_name": null,
  "port": 6379,
  "status": "RUNNING",
  "freeze_scene": null,
  "created_at": "2020-06-18T03:11:30.618Z",
  "update_at": "2020-06-18T03:11:58.937Z",
  "error_code": null,
  "user_id": "05040c67cc00256f1f28c009a1c2a0f7",
  "user_name": "dcstest",

```

```

    "maintain_begin": "02:00:00",
    "maintain_end": "06:00:00",
    "no_password_access": "true",
    "access_user": null,
    "enable_publicip": false,
    "publicip_id": null,
    "publicip_address": null,
    "enable_ssl": false,
    "service_upgrade": false,
    "service_task_id": "",
    "enterprise_project_id": "0",
    "security_group_id": null,
    "tags": null,
    "product_type": "generic",
    "cpu_type": "x86_64",
    "storage_type": "DRAM",
    "task_status": null,
    "launched_at": "2020-06-18T03:11:58.943Z",
    "is_free": null,
    "libos": false,
    "cache_mode": "ha",
    "available_zones": ["dfeb4826256b47aa828e72d7522e108c",
    "043c7e39ecb347a08dc8fcb6c35a274e"],
    "subnet_id": "e6eb5ff4-da2a-4323-bddc-c7b179f9aa1e",
    "backend_addrs": null,
    "crr_role": null,
    "cloud_service_type_code": "hws.service.type.dcs",
    "cloud_resource_type_code": "hws.resource.type.dcs3",
    "support_slow_log_flag": null,
    "ipv6": null,
    "enable_ipv6": false,
    "description": ""
  }],
  "instance_num": 2
}

```

步骤2 根据需要，记录要删除实例的instance_id。

----结束

步骤 2：批量删除实例

- 接口相关信息
URI格式：DELETE /v2/{project_id}/instances
- 调试
您可以在[API Explorer](#)中调试该接口。
- 请求示例
DELETE https://{dcs_endpoint}/v2/{project_id}/instances
{dcs_endpoint}信息，请从[地区和终端节点](#)获取。
Body:

```

{
  "instances": [
    "aabe73af-1db8-4401-b39c-c56996023989",
    "9d78d0f4-a110-4bcd-8ca3-5aec414163c1"
  ]
}

```

- 响应示例
{
 "results": [
 {
 "result": "success",
 "instance": "aabe73af-1db8-4401-b39c-c56996023989"
 }
]
}

```
},  
{  
  "result": "success",  
  "instance": "9d78d0f4-a110-4bcd-8ca3-5aec414163c1"  
}  
]  
}
```

批量删除Redis实例的参数说明请参考[批量删除实例](#)。

5 API V2（推荐）

5.1 生命周期管理

5.1.1 创建缓存实例

功能介绍

创建缓存实例，该接口创建的缓存实例支持按需和包周期两种计费方式。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances

表 5-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-2 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	参数解释： 由英文字符开头，只能由英文字母、数字、中划线和下划线组成。 约束限制： 创建单个实例时，名称长度为4到64位的字符串。批量创建实例时，名称长度为4到56位的字符串，且实例名称格式为“自定义名称-n”，其中n从000开始，依次递增。例如，批量创建两个实例，自定义名称为dcs_demo，则两个实例的名称为dcs_demo-000和dcs_demo-001。 取值范围： 不涉及。 默认取值： 不涉及。
engine	是	String	参数解释： 缓存引擎：Redis。 约束限制： 不涉及。 取值范围： Redis。 默认取值： 不涉及。
engine_version	否	String	参数解释： Redis缓存引擎。 约束限制： 不涉及。 取值范围： 4.0、5.0、6.0、7.0。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
capacity	是	Float	参数解释：缓存容量（G Byte）。约束限制：不涉及。取值范围： - Redis4.0、Redis5.0和Redis6.0基础版：单机和主备类型实例取值：0.125、0.25、0.5、1、2、4、8、16、24、32、48、64。集群实例规格支持：4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024。读写分离实例规格：1、2、4、8、16、32、64。 - Redis 7.0：单机和主备类型实例取值：0.125、0.25、0.5、1、2、4、8、16、24、32、48、64。Cluster集群实例规格支持：4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024。 默认取值： 不涉及。
spec_code	是	String	参数解释： 产品规格编码。具体查询方法如下： - 方法一：登录分布式缓存的控制台，在创建缓存实例页面，查找对应的实例规格名称。 - 方法二：调用查询产品规格接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
az_codes	是	Array of strings	参数解释： 创建缓存节点到指定且有资源的可用区Code。具体查询方法，请参考查询可用区信息，在查询时，请注意查看该可用区是否有资源。 如果是创建主备、Proxy集群、Cluster集群实例，支持跨可用区部署，可以为备节点指定备可用区。在为节点指定可用区时，用逗号分隔开，具体请查看示例。 约束限制： 不涉及。
vpc_id	是	String	参数解释： 缓存实例所在的虚拟私有云ID。 获取方法如下： • 方法1：登录虚拟私有云服务的控制台界面，在虚拟私有云的详情页面查找VPC ID。 • 方法2：通过虚拟私有云服务的查询VPC列表API接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
subnet_id	是	String	参数解释： 子网的网络ID。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，单击VPC下的子网，进入子网详情页面，查找网络ID。 - 方法2：通过虚拟私有云服务的查询子网列表API接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
security_group_id	否	String	参数解释： 指定实例所属的安全组。 当engine为Redis且engine_version为4.0及以上版本基础版实例时，该参数为可选。Redis4.0及以上版本基础版实例不支持安全组控制访问（会设置为默认参数值securityGroupId），只支持白名单控制。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，在安全组的详情页面查找安全组ID。 - 方法2：通过虚拟私有云服务的查询安全组列表API接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
publicip_id	否	String	参数解释： Redis缓存实例绑定的弹性IP地址的id。 如果开启了公网访问功能（即enable_publicip为true），该字段为必选。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
enterprise_project_id	否	String	参数解释： 企业项目ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
description	否	String	参数解释： 实例的描述信息。 "\"在json报文中属于特殊字符，如果参数值中需要显示\"或者\"字符，请在字符前增加转义字符\\，比如\\或者\"。 约束限制： 长度不超过1024的字符串。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
enable_ssl	否	Boolean	参数解释： Redis 3.0缓存实例开启公网访问功能时，是否选择支持ssl。该参数仅针对Redis 3.0实例，其他版本不涉及。 约束限制： 长度不超过1024的字符串。 取值范围： • true：开启 • false：不开启 默认取值： false。
enable_ipv6	否	Boolean	参数解释： 是否开启ipv4/ipv6双栈访问，请注意：子网需要同步开启ipv6。 约束限制： 长度不超过1024的字符串。 取值范围： true：开启ipv4/ipv6双栈访问。 false：不开启。 默认取值： false。
private_ip	否	String	参数解释： 创建缓存实例手动指定的IP地址。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_num	否	Integer	参数解释： 表示批量创建缓存实例时，创建的实例个数。 约束限制： 不涉及。 取值范围： 1-100 默认取值： 1
maintain_begin	否	String	参数解释： 维护时间窗开始时间。 约束限制： 必须是UTC时间，格式为HH:mm:ss - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为18:00:00、19:00:00等整点时间。 - 该参数不能单独为空，若该值为空，则结束时间也为空。 取值范围： 不涉及。 默认取值： 18:00:00。

参数	是否必选	参数类型	描述
maintain_end	否	String	参数解释： 维护时间窗结束时间。 约束限制： 必须是UTC时间，格式为HH:mm:ss - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为18:00:00、19:00:00等整点时间。 - 该参数不能单独为空，若该值为空，则结束时间也为空。 取值范围： 不涉及。 默认取值： 19:00:00。

参数	是否必选	参数类型	描述
password	否	String	参数解释： 缓存实例的认证信息。 当“no_password_access”配置为“false”或未配置时，请求消息中须包含password参数。 约束限制： Redis类型的缓存实例密码复杂度要求： • 输入长度为8到64位的字符串。 • 不能使用正序或逆序用户名作为密码。 • 必须包含如下四种字符中的三种组合： - 小写字母 - 大写字母 - 数字 - 特殊字符包括（`~!@#\$%^&*()-_+=\ {}";<.>/?`） 取值范围： 不涉及。 默认取值： 不涉及。
no_password_access	否	Boolean	参数解释： 是否允许免密码访问缓存实例。 约束限制： 不涉及。 取值范围： • true：该实例无需密码即可访问。 • false：该实例必须通过密码认证才能访问。 默认取值： false。

参数	是否必选	参数类型	描述
bss_param	否	BssParam object	参数解释： 表示计费模式的相关参数。置空时，默认计费模式为按需计费；否则是包周期方式。 约束限制： 包周期实例必填参数。
instance_backup_policy	否	BackupPolicy object	参数解释： 备份策略。 约束限制： 实例类型为主备和集群时支持。
tags	否	Array of ResourceTag objects	参数解释： 实例标签键值。 约束限制： 不涉及。
access_user	否	String	参数解释： 缓存类型为Redis时，不需要设置，保持为空即可。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
enable_publicip	否	Boolean	参数解释： Redis 3.0缓存实例是否开启公网访问功能。 约束限制： 不涉及。 取值范围： • true：开启 • false：不开启 默认取值： false

参数	是否必选	参数类型	描述
port	否	Integer	参数解释： 实例自定义端口。 约束限制： 只有创建Redis4.0及以上版本的基础版实例才支持自定义端口，企业版实例不支持自定义端口，默认端口为6379。 创建Redis实例时，如果没发送该参数或该参数为空，表示实例使用默认端口6379。如果自定义端口，端口范围为1~65535的任意数字。 取值范围： 1~65535 默认取值： 6379
rename_commands	否	Object	参数解释： 支持自定义重命名高危命令。创建Redis实例，如果没发送该参数或该参数为空，表示没有需要重命名的命令。 约束限制： 仅创建Redis4.0及以上版本的实例支持重命名高危命令。 取值范围： 当前支持重命名的高危命令有command、keys、flushdb、flushall、hgetall、scan、hscan、sscan、和zscan，Proxy集群实例还支持dbsize和dbstats命令重命名，其他命令暂不支持重命名。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
template_id	否	String	参数解释： 参数模板ID，不配置时使用默认参数模板，如需指定自定义参数模板，请参见 查询参数模板列表 获取模板ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
spec_param	否	SpecParam object	参数解释： 规格参数，创建spec_type为node类型的线性规格需要填写该参数。 约束限制： 不涉及。
backup_id	否	String	参数解释： 备份ID，恢复实例时使用。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
recover_instance_id	否	String	参数解释： 从回收站恢复实例ID，恢复实例时使用。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-3 BssParam

参数	是否必选	参数类型	描述
is_auto_renew	否	String	参数解释： 当选择包年包月时，该字段为必选，表示是否自动续订资源。 约束限制： 如果设置为自动续订，到期后，会自动续订一个月（自动续订时间后续可能会变化），详情可联系客服咨询。 取值范围： ● false：不自动续订。 ● true：自动续订。 默认取值： false
charging_mode	是	String	参数解释： 功能说明：付费方式（预付费、按需付费；预付费，即包周期付费）。 后付费的场景下，bss_param参数的其他字段都会被忽略。 约束限制： 不涉及。 取值范围： ● prePaid：预付费，即包年包月； ● postPaid：后付费，即按需付费； 默认取值： postPaid

参数	是否必选	参数类型	描述
is_auto_pay	否	String	参数解释： 功能说明：下单订购后，是否自动从客户的账户中支付；默认是“不自动支付”。 约束限制： 自动支付时，只能使用账户的现金支付；如果要使用代金券，请选择不自动支付，然后在用户费用中心，选择代金券支付。 如果包周期购买，没有设置成自动支付，即设置为false时，在变更规格或创建实例之后，用户必须在“费用中心 > 我的订单”，完成订单支付，否则订单一直在支付中，实例没有变更或者创建成功。 取值范围： • true：是（自动支付，从账户余额自动扣费） • false：否（默认值，只提交订单不支付，需要客户手动去支付） 默认取值： false
period_type	否	String	参数解释： 订购资源的周期类型。 约束限制： 当选择包年包月时，该字段为必选。 取值范围： • month：表示包月 • year：表示包年 默认取值： 不涉及。

参数	是否必选	参数类型	描述
period_num	否	Integer	参数解释： 订购周期数。 约束限制： 当选择包年包月时，该字段为必选。 取值范围： (后续会随运营策略变化) • period_type为month时，为 [1,9] • period_type为year时，为 [1,3] 默认取值： 不涉及。

表 5-4 BackupPolicy

参数	是否必选	参数类型	描述
backup_type	是	String	参数解释： 备份类型。 约束限制： 不涉及。 取值范围： • auto：自动备份。 • manual：手动备份。 默认取值： 不涉及。
save_days	否	Integer	参数解释： 当backup_type设置为auto时，该参数为必填。 保留天数，单位：天，取值范围：1-7。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
periodical_backup_plan	否	BackupPlanRequest object	参数解释： 备份计划。 约束限制： 当backup_type设置为auto时，该参数为必填。由自动切换为手动时，保留自动备份策略信息。

表 5-5 BackupPlanRequest

参数	是否必选	参数类型	描述
timezone_offset	否	String	参数解释： 备份的时区。该参数已废弃，没有实际作用。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
backup_at	是	Array of integers	参数解释： 每周的周几开始备份。由自动切换为手动时，保留自动备份策略信息。取值1-7，分别表示星期一到星期日。 约束限制： 不涉及。
period_type	是	String	参数解释： 备份周期类型，目前支持“weekly”。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
begin_at	是	String	参数解释： 备份开始执行时间，该时间为UTC时间，例如 “00:00-01:00”代表UTC时间0点开始执行备份。只能是整点时间段，间隔时间限定为一个小时。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-6 ResourceTag

参数	是否必选	参数类型	描述
key	是	String	参数解释： 标签键。 约束限制： - 不能为空。 - 对于同一资源键值唯一。 - 长度不超过36个字符。 - 标签的键必须唯一且输入不能为空。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
value	否	String	参数解释： 标签值。 约束限制： • action为create时必选。 action为delete时非必选。 • 长度不超过43个字符。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-7 SpecParam

参数	是否必选	参数类型	描述
sharding_count	否	Integer	参数解释： 实例分片数。 约束限制： 不涉及。 取值范围： 1-128。 默认取值： 不涉及。
replica_count	否	Integer	参数解释： 实例副本数。 约束限制： 不涉及。 取值范围： 1-10 默认取值： 不涉及。

参数	是否必选	参数类型	描述
cache_mode	否	String	参数解释： 缓存实例类型。 约束限制： 不涉及。 取值范围： • single：表示单机实例。 • ha：表示主备实例。 • cluster：表示cluster集群实例。 • proxy：表示Proxy集群实例。 • ha_rw_split：表示读写分离实例。 默认取值： 不涉及。

响应参数

状态码：200

表 5-8 响应 Body 参数

参数	参数类型	描述
order_id	String	参数解释： 订单ID，仅在创建包周期实例时返回。 取值范围： 不涉及
instances	Array of Instances objects	参数解释： 缓存实例ID和名称，如果批量创建实例，则会返回多个。

表 5-9 Instances

参数	参数类型	描述
instance_id	String	参数解释： 缓存实例ID。 取值范围： 不涉及。

参数	参数类型	描述
instance_name	String	参数解释： 缓存实例名称。 取值范围： 不涉及。

状态码：400

表 5-10 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-11 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-12 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-13 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-14 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- 创建一个按需计费的Redis单机实例，缓存版本是5.0，缓存容量是0.125GB。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances
{
  "name": "dcs-APITest",
  "description": "Test",
  "engine": "Redis",
  "engine_version": 5.0,
  "enterprise_project_id": 0,
  "vpc_id": "5e37b3be-950a-48e1-b498-65b63d336481",
  "security_group_id": "securityGroupId",
}
```

```
"enable_ssl" : false,
"private_ip" : "",
"az_codes" : [ "region01" ],
"capacity" : 0,
"access_user" : "",
"maintain_begin" : "18:00:00",
"maintain_end" : "19:00:00",
"password" : "*****",
"enable_publicip" : false,
"spec_code" : "redis.single.xu1.tiny.128",
"subnet_id" : "a4112635-3ec0-471c-95c3-5cf49b9533af",
"no_password_access" : false,
"template_id" : "b3ecc9c7-1133-4c47-a06a-21366e227d61"
}
```

- 创建一个按需计费的Redis主备实例，缓存容量2GB，设置了标签、自定义端口、高危命令重命名和跨可用区部署。

```
{
  "vpc_id" : "c5cd009b-9ba9-41f2-9a26-2b8c3f3ffd39",
  "description" : "Test",
  "security_group_id" : "d57fad27-e3e9-43b7-9498-0a698ab63b27",
  "port" : 4040,
  "maintain_end" : "19:00:00",
  "enable_ssl" : false,
  "private_ip" : "",
  "az_codes" : [ "region01", "region02" ],
  "capacity" : 2,
  "access_user" : "",
  "instance_num" : 1,
  "maintain_begin" : "18:00:00",
  "enterprise_project_id" : 0,
  "password" : "*****",
  "enable_publicip" : false,
  "engine" : "Redis",
  "spec_code" : "redis.ha.au1.large.2",
  "name" : "dcs-demo",
  "subnet_id" : "7e95c4d6-d7af-424c-a3cf-2be10968fe81",
  "engine_version" : 4.0,
  "no_password_access" : false,
  "rename_commands" : {
    "keys" : "keys001",
    "flushdb" : "flushdb001",
    "flushall" : "flushall001"
  },
  "tags" : [ {
    "key" : "dcs001",
    "value" : 2
  }, {
    "key" : "dcs003",
    "value" : 4
  } ]
}
```

- 创建一个包周期的Redis主备实例，缓存版本是4.0，缓存容量是0.125GB，设置了备份策略和跨可用区部署。

```
{
  "publicip_id" : "",
  "vpc_id" : "5e37b3be-950a-48e1-b498-65b63d336481",
  "description" : "Test",
  "security_group_id" : "securityGroupId",
  "maintain_begin" : "18:00:00",
  "maintain_end" : "19:00:00",
  "enable_ssl" : false,
  "private_ip" : "",
  "az_codes" : [ "region01", "region02" ],
  "capacity" : 0,
  "access_user" : "",
  "enterprise_project_id" : 0,
  "password" : "*****",

```

```
"enable_publicip" : false,
"engine" : "Redis",
"spec_code" : "redis.ha.xu1.tiny.r2.128",
"name" : "dcs-APITest1",
"subnet_id" : "a4112635-3ec0-471c-95c3-5cf49b9533af",
"engine_version" : 4.0,
"no_password_access" : false,
"instance_backup_policy" : {
  "save_days" : 3,
  "backup_type" : "auto",
  "periodical_backup_plan" : {
    "begin_at" : "16:00-17:00",
    "period_type" : "weekly",
    "backup_at" : [ 1, 2, 3, 4 ]
  }
},
"bss_param" : {
  "charging_mode" : "prePaid",
  "is_auto_pay" : true,
  "period_type" : "month",
  "period_num" : 1,
  "is_auto_renew" : false
}
}
```

- 创建一个包周期的Redis Cluster集群实例，缓存版本是4.0，缓存容量是4GB，设置了备份策略。

```
{
  "publicip_id" : "",
  "vpc_id" : "5e37b3be-950a-48e1-b498-65b63d336481",
  "description" : "Test",
  "security_group_id" : "securityGroupId",
  "maintain_end" : "19:00:00",
  "enable_ssl" : false,
  "private_ip" : "",
  "az_codes" : [ "region01" ],
  "capacity" : 4,
  "access_user" : "",
  "maintain_begin" : "18:00:00",
  "enterprise_project_id" : 0,
  "password" : "*****",
  "enable_publicip" : false,
  "engine" : "Redis",
  "spec_code" : "redis.cluster.xu1.large.r2.4",
  "name" : "dcs-APITest2",
  "subnet_id" : "a4112635-3ec0-471c-95c3-5cf49b9533af",
  "engine_version" : 4.0,
  "no_password_access" : false,
  "bss_param" : {
    "charging_mode" : "prePaid",
    "is_auto_pay" : true,
    "period_type" : "month",
    "period_num" : 1,
    "is_auto_renew" : false
  },
  "instance_backup_policy" : {
    "save_days" : 3,
    "backup_type" : "auto",
    "periodical_backup_plan" : {
      "begin_at" : "16:00-17:00",
      "period_type" : "weekly",
      "backup_at" : [ 1, 2, 3, 4, 5, 6, 7 ]
    }
  }
}
```

响应示例

状态码：200

参数解释：

下发创建缓存实例任务成功。

说明

查看实例状态为RUNNING时，即缓存实例创建成功。

- 示例 1

```
{
  "instances": [ {
    "instance_id": "3c49fd6b-fc7c-419e-9644-b6cce008653f",
    "instance_name": "dcs-test005"
  } ]
}
```

- 示例 2

```
{
  "order_id": "CS2005091653L2DAM",
  "instances": [ {
    "instance_id": "77284d69-cd51-4bf1-aa30-2e60f055800c",
    "instance_name": "dcs-APItest1"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建一个按需计费的Redis单机实例，缓存版本是5.0，缓存容量是0.125GB。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInstanceRequest request = new CreateInstanceRequest();
CreateInstanceBody body = new CreateInstanceBody();
List<String> listbodyAzCodes = new ArrayList<>();
listbodyAzCodes.add("region01");
body.withTemplateId("b3ecc9c7-1133-4c47-a06a-21366e227d61");
body.withEnablePublicip(false);
body.withAccessUser("");
body.withNoPasswordAccess(false);
body.withPassword("*****");
body.withMaintainEnd("19:00:00");
body.withMaintainBegin("18:00:00");
body.withPrivatelp("");
body.withEnableSsl(false);
body.withDescription("Test");
body.withEnterpriseProjectId("0");
body.withSecurityGroupId("securityGroupId");
body.withSubnetId("a4112635-3ec0-471c-95c3-5cf49b9533af");
body.withVpcId("5e37b3be-950a-48e1-b498-65b63d336481");
body.withAzCodes(listbodyAzCodes);
body.withSpecCode("redis.single.xu1.tiny.128");
body.withCapacity(0f);
body.withEngineVersion("5.0");
body.withEngine("Redis");
body.withName("dcs-API test");
request.withBody(body);
try {
    CreateInstanceResponse response = client.createInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 创建一个按需计费的Redis主备实例，缓存容量2GB，设置了标签、自定义端口、高危命令重命名和跨可用区部署。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();

CreateInstanceRequest request = new CreateInstanceRequest();
CreateInstanceBody body = new CreateInstanceBody();
List<ResourceTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new ResourceTag()
        .withKey("dcs001")
        .withValue("2")
);
listbodyTags.add(
    new ResourceTag()
        .withKey("dcs003")
        .withValue("4")
);
List<String> listbodyAzCodes = new ArrayList<>();
listbodyAzCodes.add("region01");
listbodyAzCodes.add("region02");
body.withRenameCommands("{\\\"keys\\\":\\\"keys001\\\",\\\"flushall\\\":\\\"flushall001\\\",\\\"flushdb\\\"\\\":\\\"flushdb001\\\"}");
body.withPort(4040);
body.withEnablePublicip(false);
body.withAccessUser("");
body.withTags(listbodyTags);
body.withNoPasswordAccess(false);
body.withPassword("*****");
body.withMaintainEnd("19:00:00");
body.withMaintainBegin("18:00:00");
body.withInstanceNum(1);
body.withPrivateIp("");
body.withEnableSsl(false);
body.withDescription("Test");
body.withEnterpriseProjectId("0");
body.withSecurityGroupId("d57fad27-e3e9-43b7-9498-0a698ab63b27");
body.withSubnetId("7e95c4d6-d7af-424c-a3cf-2be10968fe81");
body.withVpcId("c5cd009b-9ba9-41f2-9a26-2b8c3f3ffd39");
body.withAzCodes(listbodyAzCodes);
body.withSpecCode("redis.ha.au1.large.2");
body.withCapacity(2f);
body.withEngineVersion("4.0");
body.withEngine("Redis");
body.withName("dcs-demo");
request.withBody(body);
try {
    CreateInstanceResponse response = client.createInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

- 创建一个包周期的Redis主备实例，缓存版本是4.0，缓存容量是0.125GB，设置了备份策略和跨可用区部署。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateInstanceRequest request = new CreateInstanceRequest();
        CreateInstanceBody body = new CreateInstanceBody();
        List<Integer> listPeriodicalBackupPlanBackupAt = new ArrayList<>();
        listPeriodicalBackupPlanBackupAt.add(1);
        listPeriodicalBackupPlanBackupAt.add(2);
        listPeriodicalBackupPlanBackupAt.add(3);
        listPeriodicalBackupPlanBackupAt.add(4);
        BackupPlan periodicalBackupPlanInstanceBackupPolicy = new BackupPlan();
        periodicalBackupPlanInstanceBackupPolicy.withBackupAt(listPeriodicalBackupPlanBackupAt)
            .withPeriodType("weekly")
            .withBeginAt("16:00-17:00");
        BackupPolicy instanceBackupPolicybody = new BackupPolicy();
        instanceBackupPolicybody.withBackupType("auto")
            .withSaveDays(3)
            .withPeriodicalBackupPlan(periodicalBackupPlanInstanceBackupPolicy);
        BssParam bssParambody = new BssParam();
        bssParambody.withIsAutoRenew(BssParam.IsAutoRenewEnum.fromValue("false"))
            .withChargingMode(BssParam.ChargingModeEnum.fromValue("prePaid"))
            .withIsAutoPay(BssParam.IsAutoPayEnum.fromValue("true"))
            .withPeriodType(BssParam.PeriodTypeEnum.fromValue("month"))
            .withPeriodNum(1);
        List<String> listbodyAzCodes = new ArrayList<>();
        listbodyAzCodes.add("region01");
        listbodyAzCodes.add("region02");
```



```
body.withEnablePublicip(false);
body.withAccessUser("");
body.withInstanceBackupPolicy(instanceBackupPolicybody);
body.withBssParam(bssParambody);
body.withNoPasswordAccess(false);
body.withPassword("*****");
body.withMaintainEnd("19:00:00");
body.withMaintainBegin("18:00:00");
body.withPrivateIp("");
body.withEnableSsl(false);
body.withDescription("Test");
body.withEnterpriseProjectId("0");
body.withPublicIpId("");
body.withSecurityGroupId("securityGroupId");
body.withSubnetId("a4112635-3ec0-471c-95c3-5cf49b9533af");
body.withVpcId("5e37b3be-950a-48e1-b498-65b63d336481");
body.withAzCodes(listbodyAzCodes);
body.withSpecCode("redis.ha.xu1.tiny.r2.128");
body.withCapacity(0f);
body.withEngineVersion("4.0");
body.withEngine("Redis");
body.withName("dcs-APITest1");
request.withBody(body);
try {
    CreateInstanceResponse response = client.createInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 创建一个包周期的Redis Cluster集群实例，缓存版本是4.0，缓存容量是4GB，设置了备份策略。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInstanceRequest request = new CreateInstanceRequest();
CreateInstanceBody body = new CreateInstanceBody();
List<Integer> listPeriodicalBackupPlanBackupAt = new ArrayList<>();
listPeriodicalBackupPlanBackupAt.add(1);
listPeriodicalBackupPlanBackupAt.add(2);
listPeriodicalBackupPlanBackupAt.add(3);
listPeriodicalBackupPlanBackupAt.add(4);
listPeriodicalBackupPlanBackupAt.add(5);
listPeriodicalBackupPlanBackupAt.add(6);
listPeriodicalBackupPlanBackupAt.add(7);
BackupPlan periodicalBackupPlanInstanceBackupPolicy = new BackupPlan();
periodicalBackupPlanInstanceBackupPolicy.withBackupAt(listPeriodicalBackupPlanBackupAt)
    .withPeriodType("weekly")
    .withBeginAt("16:00-17:00");
BackupPolicy instanceBackupPolicybody = new BackupPolicy();
instanceBackupPolicybody.withBackupType("auto")
    .withSaveDays(3)
    .withPeriodicalBackupPlan(periodicalBackupPlanInstanceBackupPolicy);
BssParam bssParambody = new BssParam();
bssParambody.withIsAutoRenew(BssParam.IsAutoRenewEnum.fromValue("false"))
    .withChargingMode(BssParam.ChargingModeEnum.fromValue("prePaid"))
    .withIsAutoPay(BssParam.IsAutoPayEnum.fromValue("true"))
    .withPeriodType(BssParam.PeriodTypeEnum.fromValue("month"))
    .withPeriodNum(1);
List<String> listbodyAzCodes = new ArrayList<>();
listbodyAzCodes.add("region01");
body.withEnablePublicip(false);
body.withAccessUser("");
body.withInstanceBackupPolicy(instanceBackupPolicybody);
body.withBssParam(bssParambody);
body.withNoPasswordAccess(false);
body.withPassword("*****");
body.withMaintainEnd("19:00:00");
body.withMaintainBegin("18:00:00");
body.withPrivatelp("");
body.withEnableSsl(false);
body.withDescription("Test");
body.withEnterpriseProjectId("0");
body.withPublicipId("");
body.withSecurityGroupId("securityGroupId");
body.withSubnetId("a4112635-3ec0-471c-95c3-5cf49b9533af");
body.withVpcId("5e37b3be-950a-48e1-b498-65b63d336481");
body.withAzCodes(listbodyAzCodes);
body.withSpecCode("redis.cluster.xu1.large.r2.4");
body.withCapacity(4f);
body.withEngineVersion("4.0");
body.withEngine("Redis");
body.withName("dcs-API Test2");
request.withBody(body);
try {
    CreateInstanceResponse response = client.createInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
}
```

```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

- 创建一个按需计费的Redis单机实例，缓存版本是5.0，缓存容量是0.125GB。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceRequest()
        listAzCodesbody = [
            "region01"
        ]
        request.body = CreateInstanceBody(
            template_id="b3ecc9c7-1133-4c47-a06a-21366e227d61",
            enable_publicip=False,
            access_user="",
            no_password_access=False,
            password="*****",
            maintain_end="19:00:00",
            maintain_begin="18:00:00",
            private_ip="",
            enable_ssl=False,
            description="Test",
            enterprise_project_id="0",
            security_group_id="securityGroupId",
            subnet_id="a4112635-3ec0-471c-95c3-5cf49b9533af",
            vpc_id="5e37b3be-950a-48e1-b498-65b63d336481",
            az_codes=listAzCodesbody,
            spec_code="redis.single.xu1.tiny.128",
            capacity=0,
            engine_version="5.0",
            engine="Redis",
            name="dcs-APITest"
        )
        response = client.create_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 创建一个按需计费的Redis主备实例，缓存容量2GB，设置了标签、自定义端口、高危命令重命名和跨可用区部署。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceRequest()
        listTagsbody = [
            ResourceTag(
                key="dcs001",
                value="2"
            ),
            ResourceTag(
                key="dcs003",
                value="4"
            )
        ]
        listAzCodesbody = [
            "region01",
            "region02"
        ]
        request.body = CreateInstanceBody(
            rename_commands="{\"keys\":\"keys001\\\", \"flushall\":\"flushall001\\\", \"flushdb\":"
            "\"flushdb001\\\"}",
            port=4040,
            enable_publicip=False,
            access_user="",
            tags=listTagsbody,
            no_password_access=False,
            password="*****",
            maintain_end="19:00:00",
            maintain_begin="18:00:00",
            instance_num=1,
            private_ip="",
            enable_ssl=False,
            description="Test",
            enterprise_project_id="0",
            security_group_id="d57fad27-e3e9-43b7-9498-0a698ab63b27",
            subnet_id="7e95c4d6-d7af-424c-a3cf-2be10968fe81",
            vpc_id="c5cd009b-9ba9-41f2-9a26-2b8c3f3ffd39",
            az_codes=listAzCodesbody,
            spec_code="redis.ha.au1.large.2",
            capacity=2,
            engine_version="4.0",
            engine="Redis",
            name="dcs-demo"
```

```
)  
response = client.create_instance(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- 创建一个包周期的Redis主备实例，缓存版本是4.0，缓存容量是0.125GB，设置了备份策略和跨可用区部署。

```
# coding: utf-8
```

```
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdcs.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    # environment variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before  
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
    # environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = DcsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = CreateInstanceRequest()  
        listBackupAtPeriodicalBackupPlan = [  
            1,  
            2,  
            3,  
            4  
        ]  
        periodicalBackupPlanInstanceBackupPolicy = BackupPlan(  
            backup_at=listBackupAtPeriodicalBackupPlan,  
            period_type="weekly",  
            begin_at="16:00-17:00"  
        )  
        instanceBackupPolicybody = BackupPolicy(  
            backup_type="auto",  
            save_days=3,  
            periodical_backup_plan=periodicalBackupPlanInstanceBackupPolicy  
        )  
        bssParambody = BssParam(  
            is_auto_renew="false",  
            charging_mode="prePaid",  
            is_auto_pay="true",  
            period_type="month",  
            period_num=1  
        )  
        listAzCodesbody = [  
            "region01",  
            "region02"  
        ]  
        request.body = CreateInstanceBody(  
            enable_publicip=False,  
            access_user="",
```

```

        instance_backup_policy=instanceBackupPolicybody,
        bss_param=bssParambody,
        no_password_access=False,
        password="*****",
        maintain_end="19:00:00",
        maintain_begin="18:00:00",
        private_ip="",
        enable_ssl=False,
        description="Test",
        enterprise_project_id="0",
        publicip_id="",
        security_group_id="securityGroupId",
        subnet_id="a4112635-3ec0-471c-95c3-5cf49b9533af",
        vpc_id="5e37b3be-950a-48e1-b498-65b63d336481",
        az_codes=listAzCodesbody,
        spec_code="redis.ha.xu1.tiny.r2.128",
        capacity=0,
        engine_version="4.0",
        engine="Redis",
        name="dcs-APITest1"
    )
    response = client.create_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

- 创建一个包周期的Redis Cluster集群实例，缓存版本是4.0，缓存容量是4GB，设置了备份策略。

coding: utf-8

```

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceRequest()
        listBackupAtPeriodicalBackupPlan = [
            1,
            2,
            3,
            4,
            5,
            6,
            7
        ]
        periodicalBackupPlanInstanceBackupPolicy = BackupPlan(

```

```

        backup_at=listBackupAtPeriodicalBackupPlan,
        period_type="weekly",
        begin_at="16:00-17:00"
    )
    instanceBackupPolicybody = BackupPolicy(
        backup_type="auto",
        save_days=3,
        periodical_backup_plan=periodicalBackupPlanInstanceBackupPolicy
    )
    bssParambody = BssParam(
        is_auto_renew="false",
        charging_mode="prePaid",
        is_auto_pay="true",
        period_type="month",
        period_num=1
    )
    listAzCodesbody = [
        "region01"
    ]
    request.body = CreateInstanceBody(
        enable_publicip=False,
        access_user="",
        instance_backup_policy=instanceBackupPolicybody,
        bss_param=bssParambody,
        no_password_access=False,
        password="*****",
        maintain_end="19:00:00",
        maintain_begin="18:00:00",
        private_ip="",
        enable_ssl=False,
        description="Test",
        enterprise_project_id="0",
        publicip_id="",
        security_group_id="securityGroupId",
        subnet_id="a4112635-3ec0-471c-95c3-5cf49b9533af",
        vpc_id="5e37b3be-950a-48e1-b498-65b63d336481",
        az_codes=listAzCodesbody,
        spec_code="redis.cluster.xu1.large.r2.4",
        capacity=4,
        engine_version="4.0",
        engine="Redis",
        name="dcs-APITest2"
    )
    response = client.create_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

- 创建一个按需计费的Redis单机实例，缓存版本是5.0，缓存容量是0.125GB。

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before

```

running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateInstanceRequest{}
var listAzCodesbody = []string{
    "region01",
}
templateIdCreateInstanceBody := "b3ecc9c7-1133-4c47-a06a-21366e227d61"
enablePublicIpCreateInstanceBody := false
accessUserCreateInstanceBody := ""
noPasswordAccessCreateInstanceBody := false
passwordCreateInstanceBody := "*****"
maintainEndCreateInstanceBody := "19:00:00"
maintainBeginCreateInstanceBody := "18:00:00"
privateIpCreateInstanceBody := ""
enableSslCreateInstanceBody := false
descriptionCreateInstanceBody := "Test"
enterpriseProjectIdCreateInstanceBody := "0"
securityGroupIdCreateInstanceBody := "securityGroupId"
engineVersionCreateInstanceBody := "5.0"
request.Body = &model.CreateInstanceBody{
    TemplateId: &templateIdCreateInstanceBody,
    EnablePublicIp: &enablePublicIpCreateInstanceBody,
    AccessUser: &accessUserCreateInstanceBody,
    NoPasswordAccess: &noPasswordAccessCreateInstanceBody,
    Password: &passwordCreateInstanceBody,
    MaintainEnd: &maintainEndCreateInstanceBody,
    MaintainBegin: &maintainBeginCreateInstanceBody,
    PrivateIp: &privateIpCreateInstanceBody,
    EnableSsl: &enableSslCreateInstanceBody,
    Description: &descriptionCreateInstanceBody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceBody,
    SecurityGroupId: &securityGroupIdCreateInstanceBody,
    SubnetId: "a4112635-3ec0-471c-95c3-5cf49b9533af",
    VpId: "5e37b3be-950a-48e1-b498-65b63d336481",
    AzCodes: listAzCodesbody,
    SpecCode: "redis.single.xu1.tiny.128",
    Capacity: float32(0),
    EngineVersion: &engineVersionCreateInstanceBody,
    Engine: "Redis",
    Name: "dcs-APITest",
}
response, err := client.CreateInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 创建一个按需计费的Redis主备实例，缓存容量2GB，设置了标签、自定义端口、高危命令重命名和跨可用区部署。

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceRequest{}
    valueTags := "2"
    valueTags1 := "4"
    var listTagsbody = []model.ResourceTag{
        {
            Key: "dcs001",
            Value: &valueTags,
        },
        {
            Key: "dcs003",
            Value: &valueTags1,
        },
    }
    var listAzCodesbody = []string{
        "region01",
        "region02",
    }
    var renameCommandsCreateInstanceBody interface{} = "{ \"keys\": \"keys001\", \"flushall\": \"flushall001\", \"flushdb\": \"flushdb001\" }"
    portCreateInstanceBody := int32(4040)
    enablePublicipCreateInstanceBody := false
    accessUserCreateInstanceBody := ""
    noPasswordAccessCreateInstanceBody := false
    passwordCreateInstanceBody := "*****"
    maintainEndCreateInstanceBody := "19:00:00"
    maintainBeginCreateInstanceBody := "18:00:00"
    instanceNumCreateInstanceBody := int32(1)
    privateIpCreateInstanceBody := ""
    enableSslCreateInstanceBody := false
    descriptionCreateInstanceBody := "Test"
    enterpriseProjectIdCreateInstanceBody := "0"
    securityGroupIdCreateInstanceBody := "d57fad27-e3e9-43b7-9498-0a698ab63b27"
    engineVersionCreateInstanceBody := "4.0"
    request.Body = &model.CreateInstanceBody{
        RenameCommands: &renameCommandsCreateInstanceBody,
        Port: &portCreateInstanceBody,
        EnablePublicip: &enablePublicipCreateInstanceBody,
        AccessUser: &accessUserCreateInstanceBody,
    }
}
```

```
Tags: &listTagsbody,
NoPasswordAccess: &noPasswordAccessCreateInstanceBody,
Password: &passwordCreateInstanceBody,
MaintainEnd: &maintainEndCreateInstanceBody,
MaintainBegin: &maintainBeginCreateInstanceBody,
InstanceNum: &instanceNumCreateInstanceBody,
Privatelp: &privatelpCreateInstanceBody,
EnableSsl: &enableSslCreateInstanceBody,
Description: &descriptionCreateInstanceBody,
EnterpriseProjectId: &enterpriseProjectIdCreateInstanceBody,
SecurityGroupId: &securityGroupIdCreateInstanceBody,
SubnetId: "7e95c4d6-d7af-424c-a3cf-2be10968fe81",
VpcId: "c5cd009b-9ba9-41f2-9a26-2b8c3f3ffd39",
AzCodes: listAzCodesbody,
SpecCode: "redis.ha.au1.large.2",
Capacity: float32(2),
EngineVersion: &engineVersionCreateInstanceBody,
Engine: "Redis",
Name: "dcs-demo",
}
response, err := client.CreateInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 创建一个包周期的Redis主备实例，缓存版本是4.0，缓存容量是0.125GB，设置了备份策略和跨可用区部署。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceRequest{}
    var listBackupAtPeriodicalBackupPlan = []int32{
        int32(1),
        int32(2),
        int32(3),
        int32(4),
    }
```

```

}
periodicalBackupPlanInstanceBackupPolicy := &model.BackupPlan{
    BackupAt: listBackupAtPeriodicalBackupPlan,
    PeriodType: "weekly",
    BeginAt: "16:00-17:00",
}
saveDaysInstanceBackupPolicy:= int32(3)
instanceBackupPolicybody := &model.BackupPolicy{
    BackupType: "auto",
    SaveDays: &saveDaysInstanceBackupPolicy,
    PeriodicalBackupPlan: periodicalBackupPlanInstanceBackupPolicy,
}
isAutoRenewBssParam:= model.GetBssParamIsAutoRenewEnum().FALSE
isAutoPayBssParam:= model.GetBssParamIsAutoPayEnum().TRUE
periodTypeBssParam:= model.GetBssParamPeriodTypeEnum().MONTH
periodNumBssParam:= int32(1)
bssParambody := &model.BssParam{
    IsAutoRenew: &isAutoRenewBssParam,
    ChargingMode: model.GetBssParamChargingModeEnum().PRE_PAID,
    IsAutoPay: &isAutoPayBssParam,
    PeriodType: &periodTypeBssParam,
    PeriodNum: &periodNumBssParam,
}
var listAzCodesbody = []string{
    "region01",
    "region02",
}
enablePublicipCreateInstanceBody:= false
accessUserCreateInstanceBody:= ""
noPasswordAccessCreateInstanceBody:= false
passwordCreateInstanceBody:= "*****"
maintainEndCreateInstanceBody:= "19:00:00"
maintainBeginCreateInstanceBody:= "18:00:00"
privatelpCreateInstanceBody:= ""
enableSslCreateInstanceBody:= false
descriptionCreateInstanceBody:= "Test"
enterpriseProjectIdCreateInstanceBody:= "0"
publicipIdCreateInstanceBody:= ""
securityGroupIdCreateInstanceBody:= "securityGroupId"
engineVersionCreateInstanceBody:= "4.0"
request.Body = &model.CreateInstanceBody{
    EnablePublicip: &enablePublicipCreateInstanceBody,
    AccessUser: &accessUserCreateInstanceBody,
    InstanceBackupPolicy: instanceBackupPolicybody,
    BssParam: bssParambody,
    NoPasswordAccess: &noPasswordAccessCreateInstanceBody,
    Password: &passwordCreateInstanceBody,
    MaintainEnd: &maintainEndCreateInstanceBody,
    MaintainBegin: &maintainBeginCreateInstanceBody,
    Privatelp: &privatelpCreateInstanceBody,
    EnableSsl: &enableSslCreateInstanceBody,
    Description: &descriptionCreateInstanceBody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceBody,
    PublicipId: &publicipIdCreateInstanceBody,
    SecurityGroupId: &securityGroupIdCreateInstanceBody,
    SubnetId: "a4112635-3ec0-471c-95c3-5cf49b9533af",
    VpcId: "5e37b3be-950a-48e1-b498-65b63d336481",
    AzCodes: listAzCodesbody,
    SpecCode: "redis.ha.xu1.tiny.r2.128",
    Capacity: float32(0),
    EngineVersion: &engineVersionCreateInstanceBody,
    Engine: "Redis",
    Name: "dcs-APITest1",
}
response, err := client.CreateInstance(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}

```

- 创建一个包周期的Redis Cluster集群实例，缓存版本是4.0，缓存容量是4GB，设置了备份策略。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceRequest{}
    var listBackupAtPeriodicalBackupPlan = []int32{
        int32(1),
        int32(2),
        int32(3),
        int32(4),
        int32(5),
        int32(6),
        int32(7),
    }
    periodicalBackupPlanInstanceBackupPolicy := &model.BackupPlan{
        BackupAt: listBackupAtPeriodicalBackupPlan,
        PeriodType: "weekly",
        BeginAt: "16:00-17:00",
    }
    saveDaysInstanceBackupPolicy := int32(3)
    instanceBackupPolicybody := &model.BackupPolicy{
        BackupType: "auto",
        SaveDays: &saveDaysInstanceBackupPolicy,
        PeriodicalBackupPlan: periodicalBackupPlanInstanceBackupPolicy,
    }
    isAutoRenewBssParam := model.GetBssParamsAutoRenewEnum().FALSE
    isAutoPayBssParam := model.GetBssParamsAutoPayEnum().TRUE
    periodTypeBssParam := model.GetBssParamPeriodTypeEnum().MONTH
    periodNumBssParam := int32(1)
    bssParambody := &model.BssParam{
        IsAutoRenew: &isAutoRenewBssParam,
        ChargingMode: model.GetBssParamChargingModeEnum().PRE_PAID,
        IsAutoPay: &isAutoPayBssParam,
        PeriodType: &periodTypeBssParam,
        PeriodNum: &periodNumBssParam,
    }
```

```
}
var listAzCodesbody = []string{
    "region01",
}
enablePublicipCreateInstanceBody:= false
accessUserCreateInstanceBody:= ""
noPasswordAccessCreateInstanceBody:= false
passwordCreateInstanceBody:= "*****"
maintainEndCreateInstanceBody:= "19:00:00"
maintainBeginCreateInstanceBody:= "18:00:00"
privateIpCreateInstanceBody:= ""
enableSslCreateInstanceBody:= false
descriptionCreateInstanceBody:= "Test"
enterpriseProjectIdCreateInstanceBody:= "0"
publicIpIdCreateInstanceBody:= ""
securityGroupIdCreateInstanceBody:= "securityGroupId"
engineVersionCreateInstanceBody:= "4.0"
request.Body = &model.CreateInstanceBody{
    EnablePublicip: &enablePublicipCreateInstanceBody,
    AccessUser: &accessUserCreateInstanceBody,
    InstanceBackupPolicy: instanceBackupPolicybody,
    BssParam: bssParambody,
    NoPasswordAccess: &noPasswordAccessCreateInstanceBody,
    Password: &passwordCreateInstanceBody,
    MaintainEnd: &maintainEndCreateInstanceBody,
    MaintainBegin: &maintainBeginCreateInstanceBody,
    PrivateIp: &privateIpCreateInstanceBody,
    EnableSsl: &enableSslCreateInstanceBody,
    Description: &descriptionCreateInstanceBody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceBody,
    PublicIpId: &publicIpIdCreateInstanceBody,
    SecurityGroupId: &securityGroupIdCreateInstanceBody,
    SubnetId: "a4112635-3ec0-471c-95c3-5cf49b9533af",
    VpcId: "5e37b3be-950a-48e1-b498-65b63d336481",
    AzCodes: listAzCodesbody,
    SpecCode: "redis.cluster.xu1.large.r2.4",
    Capacity: float32(4),
    EngineVersion: &engineVersionCreateInstanceBody,
    Engine: "Redis",
    Name: "dcs-APITest2",
}
response, err := client.CreateInstance(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下发创建缓存实例任务成功。 说明 查看实例状态为RUNNING时，即缓存实例创建成功。
400	非法请求。
401	认证信息非法。

状态码	描述
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.2 查询所有实例列表

功能介绍

查询租户的缓存实例列表，支持按照条件查询。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances

表 5-15 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-16 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
include_failur e	否	String	参数解释： 是否返回创建失败的实例数。 约束限制： 不涉及。 取值范围： • true：返回创建失败的实例数。 • false：不返回创建失败的实例数。 默认取值： false
include_delete	否	String	参数解释： 是否返回已删除的实例数。 约束限制： 不涉及。 取值范围： • true：返回已删除的实例数。 • false：不返回已删除的实例数。 默认取值： false

参数	是否必选	参数类型	描述
name	否	String	参数解释： 实例名称。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10
status	否	String	参数解释： 实例状态。 约束限制： 不涉及。 取值范围： 详细状态说明见 缓存实例状态说明 。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
name_equal	否	String	参数解释： 按照实例名称进行精确匹配查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
tags	否	String	参数解释： 根据实例标签键值对进行查询。 {key}表示标签键，{value}表示标签值。 约束限制： 如果同时使用多个标签键值对进行查询，中间使用逗号分隔开，表示查询同时包含指定标签键值对的实例。 取值范围： 不涉及。 默认取值： 不涉及。
ip	否	String	参数解释： 连接缓存实例的IP地址，如192.168.7.146。可以通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
capacity	否	String	参数解释： 缓存实例的容量，单位：GB。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
is_recycle	否	Boolean	参数解释： 是否查询回收站的实例。 约束限制： 不涉及。 取值范围： • true：返回回收站的实例。 • false：返回缓存实例列表。 默认取值： false

请求参数

无

响应参数

状态码：200

表 5-17 响应 Body 参数

参数	参数类型	描述
instance_num	Integer	参数解释： 实例个数。 取值范围： 不涉及。
instances	Array of InstanceListInfo objects	参数解释： 实例的详情数组。 取值范围： 不涉及。

表 5-18 InstanceListInfo

参数	参数类型	描述
publicip_id	String	参数解释： Redis缓存实例绑定的弹性IP地址的ID。 如果未开启公网访问功能，该字段值为null。 取值范围： 不涉及。
vpc_name	String	参数解释： VPC的名称。 取值范围： 不涉及。
charging_mode	Integer	参数解释： 计费模式。 取值范围： 0：按需计费。 1：包年/包月计费。
vpc_id	String	参数解释： Redis缓存实例所在的虚拟私有云ID。 取值范围： 不涉及。
subnet_id	String	参数解释： Redis缓存实例绑定的子网ID。 取值范围： 不涉及。
security_group_id	String	参数解释： Redis缓存实例绑定的安全组ID。 取值范围： 不涉及。
created_at	String	参数解释： 实例的创建时间。格式为： 2017-03-31T12:24:46.297Z。 取值范围： 不涉及。

参数	参数类型	描述
updated_at	String	参数解释： 实例的更新时间。格式为： 2017-03-31T19:24:46.297Z。 取值范围： 不涉及。
enable_ssl	Boolean	参数解释： Redis缓存实例开启公网访问功能时，是否选择开启SSL。 取值范围： • true：开启 • false：不开启
max_memory	Integer	参数解释： 实例的总内存大小，单位：MB。 取值范围： 不涉及。
used_memory	Integer	参数解释： 实例已使用的内存大小，单位：MB。 取值范围： 不涉及。
publicip_address	String	参数解释： Redis缓存实例绑定的弹性IP地址。 如果未开启公网访问功能，该字段值为null。 取值范围： 不涉及。
capacity	Integer	参数解释： 实例的缓存容量规格大小，单位：GB。 取值范围： 不涉及。
capacity_minor	String	参数解释： 小于1GB规格缓存实例的容量，单位：GB。 取值范围： 不涉及。

参数	参数类型	描述
order_id	String	参数解释： 订单ID，仅在创建包周期实例时返回，按需实例时此值为null。 取值范围： 不涉及。
maintain_begin	String	参数解释： 维护时间窗开始时间，为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。
maintain_end	String	参数解释： 维护时间窗结束时间，为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。
engine	String	参数解释： 缓存引擎。 取值范围： - Redis - Memcached
engine_version	String	参数解释： 缓存版本。 取值范围： Redis缓存引擎的取值范围如下： 3.0 4.0 5.0 6.0 7.0 当缓存引擎为Memcached时，取值为空。
service_upgrade	Boolean	参数解释： 实例是否存在升级任务。 取值范围： - true：存在 - false：不存在

参数	参数类型	描述
no_password_access	String	参数解释： 是否允许免密码访问缓存实例。 取值范围： • true：该实例无需密码即可访问。 • false：该实例必须通过密码认证才能访问。
service_task_id	String	参数解释： 升级任务的ID。 • 当service_upgrade为true时，为升级任务的ID。 • 当service_upgrade为false时，该参数为空。 取值范围： 不涉及。
ip	String	参数解释： 连接缓存实例的IP地址。如果是集群实例，返回多个IP地址，使用逗号分隔。如：192.168.0.1，192.168.0.2。 取值范围： 不涉及。
access_user	String	参数解释： 通过密码认证访问缓存实例的认证用户名。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_publicip	Boolean	参数解释： Redis 3.0缓存实例是否开启公网访问功能。 取值范围： • true：开启 • false：不开启

参数	参数类型	描述
port	Integer	参数解释： 连接缓存实例的端口号。 取值范围： 不涉及。
user_id	String	参数解释： 实例所在的用户ID。 取值范围： 不涉及。
user_name	String	参数解释： 实例所在的用户名。 取值范围： 不涉及。
domain_name	String	参数解释： 实例的域名。 取值范围： 不涉及。
readonly_domain_name	String	参数解释： 实例的只读域名，只有主备实例有该字段。 取值范围： 不涉及。
name	String	参数解释： 实例名称。 取值范围： 不涉及。
spec_code	String	参数解释： 产品规格编码。 取值范围： 不涉及。
status	String	参数解释： 实例状态。 取值范围： 详细状态说明见 缓存实例状态说明 。

参数	参数类型	描述
tags	Array of ResourceTags objects	参数解释： 实例标签键值。 取值范围： 不涉及。
enterprise_project_id	String	参数解释： 企业项目ID。 取值范围： 不涉及。
description	String	参数解释： 实例描述信息。 取值范围： 不涉及。
cpu_type	String	参数解释： 实例CPU类型。 取值范围： - x86_64: x86计算 - aarch64: Arm计算
az_codes	Array of strings	参数解释： 有资源的可用区编码。 取值范围： 不涉及。
features	Features object	参数解释： 实例支持特性。 取值范围： 不涉及。
sub_status	String	参数解释： 实例子状态。 取值范围： 不涉及。
deleted_time	String	参数解释： 实例的删除时间，仅回收站实例展示。 格式为：2017-03-31T19:24:46.297Z。 取值范围： 不涉及。

参数	参数类型	描述
release_time	String	参数解释： 实例在回收站的预计释放时间，仅回收站实例展示。格式为：2017-03-31T19:24:46.297Z。 取值范围： 不涉及。
available_zones	Array of strings	参数解释： 有资源的可用区ID，仅回收站实例展示。
backup_id	String	参数解释： 备份ID，仅回收站实例展示。 取值范围： 不涉及。
instance_backup_policy	InstanceBackupPolicy object	参数解释： 备份策略，实例类型为主备和集群时支持，仅回收站实例展示。 取值范围： 不涉及。
cache_mode	String	参数解释： 缓存实例类型，仅回收站实例展示。 取值范围： • single：表示单机实例。 • ha：表示主备实例。 • cluster：表示cluster集群实例。 • proxy：表示Proxy集群实例。 • ha_rw_split：表示读写分离实例。
enable_ipv6	Boolean	参数解释： 是否开启ipv4/ipv6双栈访问，仅回收站实例展示。 取值范围： true：开启ipv4/ipv6双栈访问。 false：不开启
rename_commands	RenameCommandResp object	参数解释： 命令重命名列表，仅回收站实例展示。

表 5-19 ResourceTags

参数	参数类型	描述
key	String	参数解释： 标签键。 取值范围： 不涉及。
value	String	参数解释： 标签值。 取值范围： 不涉及。

表 5-20 Features

参数	参数类型	描述
support_acl	Boolean	参数解释： 实例是否支持ACL。 取值范围： • true：是 • false：否
support_transparent_client_ip	Boolean	参数解释： 实例是否支持客户端IP透传。 取值范围： • true：是 • false：否
support_ssl	Boolean	参数解释： 实例是否支持SSL。 取值范围： • true：是 • false：否
support_acl_read_policy	Boolean	参数解释： 实例是否支持ACL只读路由策略。 取值范围： • true：是 • false：否

参数	参数类型	描述
support_audit_log	Boolean	参数解释： 实例是否支持审计日志。 取值范围： • true： 是 • false： 否

表 5-21 InstanceBackupPolicy

参数	参数类型	描述
backup_policy_id	String	参数解释： 备份策略ID。 取值范围： 不涉及。
created_at	String	参数解释： 创建时间。格式为： 2022-04-11T09:45:24.790Z。 取值范围： 不涉及。
updated_at	String	参数解释： 更新时间。格式为： 2022-04-12T02:22:03.269Z。 取值范围： 不涉及。
policy	BackupPolicyRes p object	参数解释： 备份策略。 取值范围： 不涉及。
tenant_id	String	参数解释： 租户ID。 取值范围： 不涉及。

表 5-22 BackupPolicyResp

参数	参数类型	描述
backup_type	String	参数解释： 备份类型。 取值范围： • auto：自动备份。 • manual：手动备份。
save_days	Integer	参数解释： 保留天数，单位：天。由自动备份切换为手动备份时，保留自动备份策略信息。 取值范围： 1-7
periodical_backup_plan	BackupPlan object	参数解释： 备份计划。由自动备份切换为手动备份时，保留自动备份策略信息。 取值范围： 不涉及。

表 5-23 BackupPlan

参数	参数类型	描述
timezone_offset	String	参数解释： 备份的时区。该参数已废弃，没有实际作用。 取值范围： 不涉及。
backup_at	Array of integers	参数解释： 每周的周几开始备份。由自动切换为手动时，保留自动备份策略信息。 取值范围： 1-7，分别表示星期一到星期日。
period_type	String	参数解释： 备份周期类型，目前支持“weekly”。由自动切换为手动时，保留自动备份策略信息。 取值范围： 不涉及。

参数	参数类型	描述
begin_at	String	参数解释： 备份开始执行时间，该时间为UTC时间，例如“00:00-01:00”代表UTC时间0点开始执行备份。只能是整点时间段，间隔时间限定为一个小时。由自动切换为手动时，保留自动备份策略信息。 取值范围： 不涉及。

表 5-24 RenameCommandResp

参数	参数类型	描述
command	String	参数解释： 可命令重命名的命令command。 取值范围： 不涉及。
flushall	String	参数解释： 可命令重命名的命令flushall。 取值范围： 不涉及。
flushdb	String	参数解释： 可命令重命名的命令flushdb。 取值范围： 不涉及。
hgetall	String	参数解释： 可命令重命名的命令hgetall。 取值范围： 不涉及。
keys	String	参数解释： 可命令重命名的命令keys。 取值范围： 不涉及。

状态码：400

表 5-25 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-26 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-27 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-28 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-29 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- GET https://{dcs_endpoint}/v2/{project_id}/instances?offset=0&limit=10&name={name}&status={status}&id={id}&include_failure=true&include_delete=true&name_equal=false
- 查询回收站实例。
GET https://{dcs_endpoint}/v2/{project_id}/instances?is_recycle=true

响应示例

状态码：200

查询所有实例列表成功。

```
{
  "instance_num" : 1,
  "instances" : [ {
    "publicip_id" : { },
    "vpc_name" : "dcs-beta",
    "charging_mode" : 0,
    "vpc_id" : "5e37b3be-950a-48e1-b498-65b63d336481",
    "subnet_id" : "a4112635-3ec0-471c-95c3-5cf49b9533af",
    "user_name" : "clouduser",
    "created_at" : "2022-08-29T09:00:34.335Z",
    "updated_at" : "2022-08-29T09:00:34.335Z",
    "enable_ssl" : false,
    "max_memory" : 128,
    "publicip_address" : null,
    "capacity" : 0,
    "order_id" : null,
    "maintain_begin" : "18:00:00",
    "engine" : "Redis",
    "maintain_end" : "19:00:00",
    "capacity_minor" : ".125",
    "service_upgrade" : false,
    "no_password_access" : true,
```



```
"service_task_id" : "",
"ip" : "192.168.0.195",
"used_memory" : 2,
"access_user" : null,
"instance_id" : "c830053e-d5a3-4bc0-b6d6-ca37d1ea9dd5",
"enable_publicip" : false,
"port" : 6379,
"user_id" : "d53977d1adfb49c5b025ba7d33a13fd7",
"domain_name" : "redis-c830053e-d5a3-4bc0-b6d6-ca37d1ea9dd5.dcs.**cloud.com",
"name" : "dcs-lpvo0829",
"spec_code" : "redis.ha.xu1.tiny.r2.128",
"engine_version" : "5.0",
"status" : "RUNNING",
"security_group_id" : "securityGroupId",
"enterprise_project_id" : 0,
"tags" : [ {
  "key" : 1,
  "value" : "DCS"
} ],
"az_codes" : [ "region01", "region02" ],
"description" : "实例描述备注",
"cpu_type" : "x86_64",
"features" : {
  "support_acl" : true,
  "support_transparent_client_ip" : true,
  "support_ssl" : false,
  "support_audit_log" : false
},
"sub_status" : "normal"
} ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4800",
  "error_msg" : "Invalid offset in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ListInstancesRequest request = new ListInstancesRequest();
try {
    ListInstancesResponse response = client.listInstances(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstancesRequest()
        response = client.list_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstancesRequest{}
    response, err := client.ListInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询所有实例列表成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.3 批量删除实例

功能介绍

批量删除多个缓存实例。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances

表 5-30 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-31 Query 参数

参数	是否必选	参数类型	描述
all_failure	否	Boolean	参数解释： 是否批量删除创建失败的缓存实例。取值如下： 约束限制： 不涉及。 取值范围： • true：表示删除租户所有创建失败的缓存实例，此时请求参数instances可为空； • false：表示删除instances参数数组中指定的缓存实例。 默认取值： false。

请求参数

表 5-32 请求 Body 参数

参数	是否必选	参数类型	描述
instances	否	Array of strings	参数解释： 实例的ID列表。仅当URI中参数all_failure值为“false”或者其他值时，才需要配置该参数。 约束限制： 不涉及。
is_recycle	否	Boolean	参数解释： 实例删除后是否放入回收站。 约束限制： 不涉及。 取值范围： true：表示删除实例放入回收站。 false：表示删除实例不放入回收站，直接销毁。 默认取值： false

响应参数

状态码：200

表 5-33 响应 Body 参数

参数	参数类型	描述
results	Array of BatchOpsResult objects	参数解释： 删除/重启/清空实例的结果。

表 5-34 BatchOpsResult

参数	参数类型	描述
result	String	参数解释： 操作结果。 取值范围： success：删除/重启/清空实例成功。 failed：删除/重启/清空实例失败。
instance	String	参数解释： 缓存实例ID。 取值范围： 不涉及。

状态码：400

表 5-35 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-36 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

批量删除缓存实例。

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances?all_failure={all_failure}

{
  "instances": [ "54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-dbad-4d79-9610-7163e6f8b640" ]
}
```

响应示例

状态码：200

参数解释：

批量删除缓存实例成功。

```
{
  "results": [ {
```

```
"result" : "success",
"instance" : "54602a9d-5e22-4239-9123-77e350df4a34"
}, {
"result" : "success",
"instance" : "7166cdea-dbad-4d79-9610-7163e6f8b640"
}
}]
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量删除缓存实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchDeleteInstancesRequest request = new BatchDeleteInstancesRequest();
        BatchDeleteBody body = new BatchDeleteBody();
        List<String> listbodyInstances = new ArrayList<>();
        listbodyInstances.add("54602a9d-5e22-4239-9123-77e350df4a34");
        listbodyInstances.add("7166cdea-dbad-4d79-9610-7163e6f8b640");
        body.withInstances(listbodyInstances);
        request.withBody(body);
        try {
            BatchDeleteInstancesResponse response = client.batchDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```



```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

批量删除缓存实例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteInstancesRequest()
        listInstancesbody = [
            "54602a9d-5e22-4239-9123-77e350df4a34",
            "7166cdea-dbad-4d79-9610-7163e6f8b640"
        ]
        request.body = BatchDeleteBody(
            instances=listInstancesbody
        )
        response = client.batch_delete_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

批量删除缓存实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstancesRequest{}
    var listInstancesbody = []string{
        "54602a9d-5e22-4239-9123-77e350df4a34",
        "7166cdea-dbad-4d79-9610-7163e6f8b640",
    }
    request.Body = &model.BatchDeleteBody{
        Instances: &listInstancesbody,
    }
    response, err := client.BatchDeleteInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	批量删除缓存实例成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.4 查询指定实例

功能介绍

通过实例ID查询实例的详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}

表 5-37 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-38 响应 Body 参数

参数	参数类型	描述
vpc_name	String	参数解释： VPC的名称。 取值范围： 不涉及。
charging_mode	Integer	参数解释： 付费模式。 取值范围： 0：按需计费。 1：包年/包月计费。
vpc_id	String	参数解释： Redis缓存实例所在的虚拟私有云ID。 取值范围： 不涉及。
user_name	String	参数解释： 实例所在的用户名。 取值范围： 不涉及。
created_at	String	参数解释： 完成创建时间。格式为： 2017-03-31T12:24:46.297Z。 取值范围： 不涉及。
description	String	参数解释： 实例描述。 取值范围： 不涉及。
security_group_id	String	参数解释： Redis缓存实例绑定的安全组ID。 取值范围： 不涉及。
security_group_name	String	参数解释： 租户安全组名称。 取值范围： 不涉及。

参数	参数类型	描述
max_memory	Integer	参数解释： 总内存，单位：MB。 取值范围： 不涉及。
used_memory	Integer	参数解释： 已使用的内存，单位：MB。 取值范围： 不涉及。
capacity	Integer	参数解释： 缓存实例的容量，单位：GB。 取值范围： 不涉及。
capacity_minor	String	参数解释： 小于1GB规格缓存实例的容量，单位：GB。 取值范围： 不涉及。
maintain_begin	String	参数解释： 维护时间窗开始时间，为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。
maintain_end	String	参数解释： 维护时间窗结束时间，为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。
engine	String	参数解释： 缓存实例的引擎类型。 取值范围： 不涉及。
no_password_access	String	参数解释： 是否允许免密码访问缓存实例。 取值范围： - true：该实例无需密码即可访问。 - false：该实例必须通过密码认证才能访问。

参数	参数类型	描述
ip	String	参数解释： 连接缓存实例的IP地址。如果是集群实例，返回多个IP地址，使用逗号分隔。如：192.168.0.1，192.168.0.2。 取值范围： 不涉及。
instance_backup_policy	InstanceBackupPolicy object	参数解释： 备份策略，实例类型为主备、读写分离和集群时支持。 取值范围： 不涉及。
az_codes	Array of strings	参数解释： 实例所在的可用区。返回"可用区Code"。 取值范围： 不涉及。
available_zones	Array of strings	参数解释： 实例所在的可用区ID。返回"可用区ID"。 取值范围： 不涉及。
access_user	String	参数解释： 通过密码认证访问缓存实例的认证用户名。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
port	Integer	参数解释： 连接缓存实例的端口号。 取值范围： 不涉及。
user_id	String	参数解释： 实例所在的用户ID。 取值范围： 不涉及。

参数	参数类型	描述
name	String	参数解释： 实例名称。 取值范围： 不涉及。
spec_code	String	参数解释： 产品规格编码。 取值范围： 不涉及。
subnet_id	String	参数解释： Redis缓存实例绑定的子网ID。 取值范围： 不涉及。
subnet_name	String	参数解释： 子网名称。 取值范围： 不涉及。
subnet_cidr	String	参数解释： 子网网段。 取值范围： 不涉及。
engine_version	String	参数解释： 缓存版本。 取值范围： Redis缓存引擎的取值范围如下： • 3.0 • 4.0 • 5.0 • 6.0 • 7.0 当缓存引擎为Memcached时，取值为空。
order_id	String	参数解释： 订单ID。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 缓存实例的状态。 取值范围： 详细状态说明见 缓存实例状态说明 。
domain_name	String	参数解释： 实例的域名。 取值范围： 不涉及。
readonly_domain_name	String	参数解释： 实例的只读域名，只有主备实例有该字段。 取值范围： 不涉及。
enable_publicip	Boolean	参数解释： Redis 3.0缓存实例是否开启公网访问功能。 取值范围： - true：开启 - false：不开启
publicip_id	String	参数解释： Redis缓存实例绑定的弹性IP地址的ID。 如果未开启公网访问功能，该字段值为null。 取值范围： 不涉及。
publicip_address	String	参数解释： Redis缓存实例绑定的弹性IP地址。 如果未开启公网访问功能，该字段值为null。 取值范围： 不涉及。
enable_ssl	Boolean	参数解释： Redis缓存实例开启公网访问功能时，是否选择支持SSL。 取值范围： - true：开启 - false：不开启

参数	参数类型	描述
service_upgrade	Boolean	参数解释： 实例是否存在升级任务。 取值范围： - true：存在 - false：不存在
service_task_id	String	参数解释： 升级任务的ID。 - 当service_upgrade为true时，为升级任务的ID。 - 当service_upgrade为false时，该参数为空。 取值范围： 不涉及。
enterprise_project_id	String	参数解释： 企业项目ID。 取值范围： 不涉及。
backend_addrs	String	参数解释： 集群实例的后端服务地址。 取值范围： 不涉及。
features	Features object	参数解释： 实例支持特性。 取值范围： 不涉及。
domain_name_info	DomainNameInfo object	参数解释： 实例域名重建、公网解析操作相关信息。 取值范围： 不涉及。
transparent_client_ip_enable	Boolean	参数解释： 是否开启客户端IP透传。 取值范围： 不涉及。

参数	参数类型	描述
sub_status	String	参数解释： 实例子状态。 取值范围： 不涉及。
tags	Array of ResourceTags objects	参数解释： 实例标签键值。 取值范围： 不涉及。
cpu_type	String	参数解释： 实例CPU类型。 取值范围： - x86_64: x86计算 - aarch64: Arm计算
enterprise_project_name	String	参数解释： 企业项目名称。 取值范围： 不涉及。
update_at	String	参数解释： 更新时间，形如 2022-07-06T09:32:16.502Z。 取值范围： 不涉及。
product_type	String	参数解释： 产品类型。 取值范围： - generic: 基础版。 - enterprise: 企业版。
storage_type	String	参数解释： 存储类型。 取值范围： 内存存储。
launched_at	String	参数解释： 启动时间，形如 2022-07-06T09:32:16.502Z。 取值范围： 不涉及。

参数	参数类型	描述
cache_mode	String	参数解释： 缓存实例类型。 取值范围： • single：表示单机实例。 • ha：表示主备实例。 • cluster：表示cluster集群实例。 • proxy：表示Proxy集群实例。 • ha_rw_split：表示读写分离实例。
support_slow_log_f lag	String	参数解释： 是否支持慢日志。 取值范围： • true：支持。 • false：不支持。
db_number	Integer	参数解释： 数据库数量。 取值范围： 不涉及。
replica_count	Integer	参数解释： 副本数。 取值范围： 不涉及。
sharding_count	Integer	参数解释： 集群实例分片个数。 取值范围： 不涉及。
bandwidth_info	BandwidthInfo object	参数解释： 带宽信息。 取值范围： 不涉及。
cloud_service_type _code	String	参数解释： 云服务类型编码。 取值范围： 不涉及。

参数	参数类型	描述
cloud_resource_type_code	String	参数解释: 云资源类型编码。 取值范围: 不涉及。
inquiry_spec_code	String	参数解释: 运营系统中的规格编码。 取值范围: 不涉及。
free	String	参数解释: 该参数与if_free相同，未启用，返回null。 取值范围: 不涉及。
is_free	String	参数解释: 该参数未启用，返回null。 取值范围: 不涉及。
libos	String	参数解释: 该参数未启用，返回false。 取值范围: 不涉及。
crr_role	String	参数解释: 该参数未启用，返回null。 取值范围: 不涉及。
active_active	ActiveActiveInfo object	参数解释: 实例AA多活特性，该参数未启用，返回null。 取值范围: 不涉及。
task	AnyType	参数解释: 实例当前任务。 取值范围: 不涉及。

参数	参数类型	描述
proxy_links	Array of ProxyLink objects	参数解释： 实例proxy节点链接情况，该参数未启用，返回null 取值范围： 不涉及。

表 5-39 InstanceBackupPolicy

参数	参数类型	描述
backup_policy_id	String	参数解释： 备份策略ID。 取值范围： 不涉及。
created_at	String	参数解释： 创建时间。格式为： 2022-04-11T09:45:24.790Z。 取值范围： 不涉及。
updated_at	String	参数解释： 更新时间。格式为： 2022-04-12T02:22:03.269Z。 取值范围： 不涉及。
policy	BackupPolicyResp object	参数解释： 备份策略。 取值范围： 不涉及。
tenant_id	String	参数解释： 租户ID。 取值范围： 不涉及。

表 5-40 BackupPolicyResp

参数	参数类型	描述
backup_type	String	参数解释： 备份类型。 取值范围： • auto：自动备份。 • manual：手动备份。
save_days	Integer	参数解释： 保留天数，单位：天。由自动备份切换为手动备份时，保留自动备份策略信息。 取值范围： 1-7
periodical_backup_plan	BackupPlan object	参数解释： 备份计划。由自动备份切换为手动备份时，保留自动备份策略信息。 取值范围： 不涉及。

表 5-41 BackupPlan

参数	参数类型	描述
timezone_offset	String	参数解释： 备份的时区。该参数已废弃，没有实际作用。 取值范围： 不涉及。
backup_at	Array of integers	参数解释： 每周的周几开始备份。由自动切换为手动时，保留自动备份策略信息。 取值范围： 1-7，分别表示星期一到星期日。
period_type	String	参数解释： 备份周期类型，目前支持“weekly”。由自动切换为手动时，保留自动备份策略信息。 取值范围： 不涉及。

参数	参数类型	描述
begin_at	String	参数解释： 备份开始执行时间，该时间为UTC时间，例如“00:00-01:00”代表UTC时间0点开始执行备份。只能是整点时间段，间隔时间限定为一个小时。由自动切换为手动时，保留自动备份策略信息。 取值范围： 不涉及。

表 5-42 Features

参数	参数类型	描述
support_acl	Boolean	参数解释： 实例是否支持ACL。 取值范围： • true：是 • false：否
support_transparent_client_ip	Boolean	参数解释： 实例是否支持客户端IP透传。 取值范围： • true：是 • false：否
support_ssl	Boolean	参数解释： 实例是否支持SSL。 取值范围： • true：是 • false：否
support_acl_read_policy	Boolean	参数解释： 实例是否支持ACL只读路由策略。 取值范围： • true：是 • false：否
support_audit_log	Boolean	参数解释： 实例是否支持审计日志。 取值范围： • true：是 • false：否

表 5-43 DomainNameInfo

参数	参数类型	描述
support_public_resolve	Boolean	参数解释： 是否开启公网域名解析。 取值范围： • true：开启 • false：未开启
is_latest_rules	Boolean	参数解释： 当前域名是否已为最新。 取值范围： • true：是 • false：否
zone_name	String	参数解释： 域名的区域后缀。 取值范围： 不涉及。
history_domain_names	Array of DomainNameEntity objects	参数解释： 历史域名信息。 取值范围： 不涉及。

表 5-44 DomainNameEntity

参数	参数类型	描述
domain_name	String	参数解释： 实例历史域名。 取值范围： 不涉及。
is_readonly	Boolean	参数解释： 是否只读域名。 取值范围： • true：是 • false：否

表 5-45 ResourceTags

参数	参数类型	描述
key	String	参数解释： 标签键。 取值范围： 不涉及。
value	String	参数解释： 标签值。 取值范围： 不涉及。

表 5-46 BandwidthInfo

参数	参数类型	描述
begin_time	Long	参数解释： 临时扩容开始时间。 取值范围： 不涉及。
end_time	Long	参数解释： 临时扩容结束时间。 取值范围： 不涉及。
current_time	Long	参数解释： 当前时间。 取值范围： 不涉及。
bandwidth	Integer	参数解释： 当前带宽值，单位为Mbit/s。 取值范围： 不涉及。
next_expand_time	Long	参数解释： 下一个扩容时间。 取值范围： 不涉及。

参数	参数类型	描述
expand_count	Integer	参数解释： 扩容数量。 取值范围： 不涉及。
expand_effect_time	Long	参数解释： 临时扩容时间间隔。 取值范围： 不涉及。
expand_interval_time	Long	参数解释： 下一次可以扩容间隔时间。 取值范围： 不涉及。
max_expand_count	Integer	参数解释： 最大扩容数量。 取值范围： 不涉及。
task_running	Boolean	参数解释： 任务是否运行。 取值范围： 不涉及。
assured_bandwidth	Integer	参数解释： 实例基准带宽，单位：Mbit/s。 取值范围： 不涉及。
max_bandwidth_for_node	Integer	参数解释： 节点最大带宽，单位：Mbit/s。 取值范围： 不涉及。

表 5-47 ActiveActiveInfo

参数	参数类型	描述
active_active_status	AnyType	参数解释： 实例AA多活状态。 取值范围： enabled：可以开启AA多活。 disabled：不可以开启AA多活。 opening：已开启AA多活。 closing：已关闭AA多活。
active_cluster_index	AnyType	参数解释： 实例AA集群索引。 取值范围： 不涉及。

表 5-48 ProxyLink

参数	参数类型	描述
ip	AnyType	参数解释： proxy节点ipv4。 取值范围： 不涉及。
ipv6_ip	AnyType	参数解释： proxy节点ipv6。 取值范围： 不涉及。
az_code	AnyType	参数解释： proxy节点所在可用区信息。 取值范围： 不涉及。
role	AnyType	参数解释： proxy节点角色信息。 取值范围： 不涉及。
domain_name	AnyType	参数解释： proxy节点域名信息。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}
```

响应示例

状态码：200

查询指定实例成功。

```
{
  "publicip_id" : null,
  "vpc_name" : "dcs-beta",
  "charging_mode" : 0,
  "enterprise_project_name" : "default",
  "vpc_id" : "5e37b3be-950a-48e1-b498-65b63d336481",
  "user_name" : "clouduser",
  "created_at" : "2022-08-29T08:53:47.319Z",
  "description" : null,
  "cache_mode" : "ha",
  "security_group_id" : "securityGroupId",
  "enable_ssl" : false,
  "max_memory" : 128,
  "publicip_address" : null,
  "capacity" : 0,
  "maintain_begin" : "18:00:00",
  "engine" : "Redis",
  "maintain_end" : "19:00:00",
  "capacity_minor" : ".125",
  "service_upgrade" : false,
  "subnet_name" : "dcs-beta",
  "no_password_access" : "true",
  "service_task_id" : null,
  "ip" : "192.168.0.251",
  "subnet_cidr" : "192.168.0.0/24",
  "used_memory" : 2,
  "security_group_name" : "securityGroupId",
  "instance_backup_policy" : null,
  "az_codes" : [ "xx-xxxx-xx" ],
  "available_zones" : [ "d539378ec1314c85b76fefa3f7071458" ],
  "access_user" : null,
  "enterprise_project_id" : "0",
  "instance_id" : "a4c545cd-10b2-4efc-b0ab-a9b554ca0621",
  "enable_publicip" : false,
  "port" : 6379,
  "user_id" : "d53977d1adfb49c5b025ba7d33a13fd7",
  "domain_name" : "redis-a4c545cd-10b2-4efc-b0ab-a9b554ca0621.dcs.example.com",
  "name" : "dcs-billing",
  "spec_code" : "redis.ha.xu1.tiny.r2.128",
  "subnet_id" : "a4112635-3ec0-471c-95c3-5cf49b9533af",
  "engine_version" : "5.0",
  "order_id" : null,
  "status" : "RUNNING",
  "features" : {
    "support_acl" : true,
    "support_transparent_client_ip" : true,
    "support_ssl" : false,
    "support_audit_log" : false
  },
  "domain_name_info" : {
    "support_public_resolve" : true,
    "is_latest_rules" : true,
    "zone_name" : "dcs.example.com",
    "history_domain_names" : null
  },
  "transparent_client_ip_enable" : true,
  "sub_status" : "normal",
  "cpu_type" : "x86_64",
  "cloud_service_type_code" : "hws.service.type.dcs",
}
```

```
"cloud_resource_type_code": "hws.resource.type.dcs3",
"inquiry_spec_code": "redis.ha.xu1.tiny.r2.128",
"free": null,
"is_free": null,
"libos": false,
"crr_role": null,
"active_active": null,
"task": "EXTENDING",
"proxy_links": null
}
```

状态码

状态码	描述
200	查询指定实例成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.5 删除实例

功能介绍

删除指定的缓存实例，释放该实例的所有资源。

 说明

如果是删除按需资源，请按照本章节执行，如果是删除包周期资源，即退订，请参考[退订包周期资源](#)。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}

表 5-49 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-50 请求 Body 参数

参数	是否必选	参数类型	描述
is_recycle	否	Boolean	参数解释： 实例删除后是否放入回收站。 约束限制： 不涉及。 取值范围： true：表示删除实例放入回收站。 false：表示删除实例不放入回收站，直接销毁。 默认取值： false

响应参数

状态码：204

删除实例成功。

无

请求示例

- 删除指定实例。

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}

{
  "is_recycle" : false
}
```
- 删除指定实例，并添加进回收站。

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}

{
  "is_recycle" : true
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

- 删除指定实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteSingleInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSingleInstanceRequest request = new DeleteSingleInstanceRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteSingleInstanceResponse response = client.deleteSingleInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 删除指定实例，并添加进回收站。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class DeleteSingleInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteSingleInstanceRequest request = new DeleteSingleInstanceRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteSingleInstanceResponse response = client.deleteSingleInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 删除指定实例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = DeleteSingleInstanceRequest()
    request.instance_id = "{instance_id}"
    response = client.delete_single_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 删除指定实例，并添加进回收站。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSingleInstanceRequest()
        request.instance_id = "{instance_id}"
        response = client.delete_single_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 删除指定实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
```

running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteSingleInstanceRequest{}
request.InstanceId = "{instance_id}"
response, err := client.DeleteSingleInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 删除指定实例，并添加进回收站。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSingleInstanceRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.DeleteSingleInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
    }  
  }
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除实例成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.6 修改实例信息

功能介绍

修改缓存实例的信息，可修改信息包括实例名称、描述、端口号、备份策略、维护时间窗开始和结束时间以及安全组。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}

表 5-51 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-52 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	参数解释： 实例名称。 约束限制： 由英文字符开头，只能由英文字母、数字、中划线和下划线组成，名称长度为4到64位的字符串。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
description	否	String	参数解释： 实例的描述信息。 约束限制： 长度不超过1024的字符串。 说明 \与"在json报文中属于特殊字符，如果参数值中需要显示\或者"字符，请在字符前增加转义字符\，比如\或者"。 取值范围： 不涉及。 默认取值： 不涉及。
port	否	Integer	参数解释： 实例自定义端口。 约束限制： 只有创建Redis4.0及以上版本的基础版实例才支持自定义端口，企业版实例不支持自定义端口，默认端口为6379。 创建Redis实例时，如果没发送该参数或该参数为空，表示实例使用默认端口6379。如果自定义端口，端口范围为1~65535的任意数字。 取值范围： 1~65535 默认取值： 6379
rename_commands	否	RenameCommandRequest object	参数解释： 命令重命名列表。

参数	是否必选	参数类型	描述
maintain_begin	否	String	参数解释： 维护时间窗开始时间。 约束限制： 必须是UTC时间，格式为HH:mm:ss - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为18:00:00、19:00:00等整点时间。 - 该参数不能单独为空，若该值为空，则结束时间也为空。 取值范围： 不涉及。 默认取值： 18:00:00。
maintain_end	否	String	参数解释： 维护时间窗结束时间。 约束限制： 必须是UTC时间，格式为HH:mm:ss - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为18:00:00、19:00:00等整点时间。 - 该参数不能单独为空，若该值为空，则结束时间也为空。 取值范围： 不涉及。 默认取值： 19:00:00。

参数	是否必选	参数类型	描述
security_group_id	否	String	参数解释： 指定实例所属的安全组。 当engine为Redis且engine_version为4.0及以上版本基础版实例时，该参数为可选。 Redis4.0及以上版本基础版实例不支持安全组控制访问(会设置为默认参数值securityGroupId)，只支持白名单控制。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，在安全组的详情页面查找安全组ID。 - 方法2：通过虚拟私有云服务的查询安全组列表API接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_backup_policy	否	BackupPolicy object	参数解释： 备份策略，实例类型为主备和集群时支持。

表 5-53 RenameCommandRequest

参数	是否必选	参数类型	描述
command	否	String	参数解释： 可命令重命名的命令command。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
flushall	否	String	参数解释： 可命令重命名的命令flushall。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
flushdb	否	String	参数解释： 可命令重命名的命令flushdb。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
hgetall	否	String	参数解释： 可命令重命名的命令hgetall。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
keys	否	String	参数解释： 可命令重命名的命令hgetall。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-54 BackupPolicy

参数	是否必选	参数类型	描述
backup_type	是	String	参数解释： 备份类型。 约束限制： 不涉及。 取值范围： • auto：自动备份。 • manual：手动备份。 默认取值： 不涉及。
save_days	否	Integer	参数解释： 当backup_type设置为auto时，该参数为必填。 保留天数，单位：天，取值范围：1-7。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
periodical_backup_plan	否	BackupPlanRequest object	参数解释： 备份计划。 约束限制： 当backup_type设置为auto时，该参数为必填。由自动切换为手动时，保留自动备份策略信息。

表 5-55 BackupPlanReqst

参数	是否必选	参数类型	描述
timezone_offset	否	String	参数解释： 备份的时区。该参数已废弃，没有实际作用。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
backup_at	是	Array of integers	参数解释： 每周的周几开始备份。由自动切换为手动时，保留自动备份策略信息。取值1-7，分别表示星期一到星期日。 约束限制： 不涉及。
period_type	是	String	参数解释： 备份周期类型，目前支持“weekly”。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
begin_at	是	String	参数解释： 备份开始执行时间，该时间为UTC时间，例如 “00:00-01:00”代表UTC时间0点开始执行备份。只能是整点时间段，间隔时间限定为一个小时。由自动切换为手动时，保留自动备份策略信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：204

修改缓存实例成功。

无

请求示例

修改指定实例的信息，修改实例名称为dcs002，端口为6379，设置自动备份策略。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}

{
  "name": "dcs002",
  "description": "instance description",
  "port": 6379,
  "maintain_begin": "18:00:00",
  "maintain_end": "19:00:00",
  "security_group_id": "18e9309f-f81a-4749-bb21-f74576292162",
  "instance_backup_policy": {
    "backup_type": "auto",
    "save_days": 1,
    "periodical_backup_plan": {
      "backup_at": [ "1", "2", "3", "4", "6", "7" ],
      "period_type": "weekly",
      "begin_at": "00:00-01:00"
    }
  }
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

修改指定实例的信息，修改实例名称为dcs002，端口为6379，设置自动备份策略。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        ModifyInstanceBody body = new ModifyInstanceBody();
        List<Integer> listPeriodicalBackupPlanBackupAt = new ArrayList<>();
        listPeriodicalBackupPlanBackupAt.add(1);
        listPeriodicalBackupPlanBackupAt.add(2);
        listPeriodicalBackupPlanBackupAt.add(3);
        listPeriodicalBackupPlanBackupAt.add(4);
        listPeriodicalBackupPlanBackupAt.add(6);
        listPeriodicalBackupPlanBackupAt.add(7);
        BackupPlan periodicalBackupPlanInstanceBackupPolicy = new BackupPlan();
        periodicalBackupPlanInstanceBackupPolicy.withBackupAt(listPeriodicalBackupPlanBackupAt)
            .withPeriodType("weekly")
            .withBeginAt("00:00-01:00");
        BackupPolicy instanceBackupPolicybody = new BackupPolicy();
        instanceBackupPolicybody.withBackupType("auto")
            .withSaveDays(1)
            .withPeriodicalBackupPlan(periodicalBackupPlanInstanceBackupPolicy);
        body.withInstanceBackupPolicy(instanceBackupPolicybody);
        body.withSecurityGroupId("18e9309f-f81a-4749-bb21-f74576292162");
        body.withMaintainEnd("19:00:00");
        body.withMaintainBegin("18:00:00");
        body.withPort(6379);
        body.withDescription("instance description");
        body.withName("dcs002");
```

```
request.withBody(body);
try {
    UpdateInstanceResponse response = client.updateInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

修改指定实例的信息，修改实例名称为dcs002，端口为6379，设置自动备份策略。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceRequest()
        request.instance_id = "{instance_id}"
        listBackupAtPeriodicalBackupPlan = [
            1,
            2,
            3,
            4,
            6,
            7
        ]
        periodicalBackupPlanInstanceBackupPolicy = BackupPlan(
            backup_at=listBackupAtPeriodicalBackupPlan,
            period_type="weekly",
            begin_at="00:00-01:00"
        )
        instanceBackupPolicybody = BackupPolicy(
            backup_type="auto",
            save_days=1,
            periodical_backup_plan=periodicalBackupPlanInstanceBackupPolicy
        )
        request.body = ModifyInstanceBody(
```

```
        instance_backup_policy=instanceBackupPolicybody,
        security_group_id="18e9309f-f81a-4749-bb21-f74576292162",
        maintain_end="19:00:00",
        maintain_begin="18:00:00",
        port=6379,
        description="instance description",
        name="dcs002"
    )
    response = client.update_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

修改指定实例的信息，修改实例名称为dcs002，端口为6379，设置自动备份策略。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{}
    request.InstanceId = "{instance_id}"
    var listBackupAtPeriodicalBackupPlan = []int32{
        int32(1),
        int32(2),
        int32(3),
        int32(4),
        int32(6),
        int32(7),
    }
    periodicalBackupPlanInstanceBackupPolicy := &model.BackupPlan{
        BackupAt: listBackupAtPeriodicalBackupPlan,
        PeriodType: "weekly",
        BeginAt: "00:00-01:00",
    }
    saveDaysInstanceBackupPolicy := int32(1)
    instanceBackupPolicybody := &model.BackupPolicy{
```

```
BackupType: "auto",
SaveDays: &saveDaysInstanceBackupPolicy,
PeriodicalBackupPlan: periodicalBackupPlanInstanceBackupPolicy,
}
securityGroupIdModifyInstanceBody:= "18e9309f-f81a-4749-bb21-f74576292162"
maintainEndModifyInstanceBody:= "19:00:00"
maintainBeginModifyInstanceBody:= "18:00:00"
portModifyInstanceBody:= int32(6379)
descriptionModifyInstanceBody:= "instance description"
nameModifyInstanceBody:= "dcs002"
request.Body = &model.ModifyInstanceBody{
    InstanceBackupPolicy: instanceBackupPolicybody,
    SecurityGroupId: &securityGroupIdModifyInstanceBody,
    MaintainEnd: &maintainEndModifyInstanceBody,
    MaintainBegin: &maintainBeginModifyInstanceBody,
    Port: &portModifyInstanceBody,
    Description: &descriptionModifyInstanceBody,
    Name: &nameModifyInstanceBody,
}
response, err := client.UpdateInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	修改缓存实例成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.7 变更实例规格

功能介绍

用户可以为状态为“运行中”的DCS缓存实例进行规格变更。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/resize

表 5-56 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-57 请求 Body 参数

参数	是否必选	参数类型	描述
spec_code	是	String	参数解释： 变更后的产品规格编码。具体查询方法如下： - 方法一：登录分布式缓存的控制台，在创建缓存实例页面，查找对应的实例规格名称。 - 方法二：调用查询产品规格接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
new_capacity	是	Integer	参数解释： 新的缓存实例规格，单位：GB。 约束限制： Redis4.0及以上版本：单机和主备类型实例取值：0.125、0.25、0.5、1、2、4、8、16、32、64。集群实例规格支持4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024。读写分离实例规格支持1、2、4、8、16、32、64。 Memcached：单机和主备类型实例取值：2、4、8、16、32、64。 说明：目前不支持3.0 proxy集群的缩容。该参数支持数字类型的0.125，0.25，0.5。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
bss_param	否	BssParamEntity object	参数解释： 按需计费无需填写，表示包周期计费模式的相关参数，包周期实例变更规格时需要填写。
reserved_ip	否	Array of strings	参数解释： 需要保留的节点IP。cluster集群扩容时需要填写，不填写时系统将随机删除多余的分片。
change_type	否	String	参数解释： 变更类型，Redis实例进行副本数变更时必选。 约束限制： 不涉及。 取值范围： - createReplication: 添加副本 - deleteReplication: 删除副本 - addSharding: 添加分片 - instanceType: 实例类型变更，目前仅支持Redis 4.0/5.0/6.0实例中的主备实例/读写分离实例变更为proxy集群实例、proxy集群实例变更为主备实例/读写分离实例、主备实例变更为读写分离实例。 默认取值： 不涉及。
available_zones	否	Array of strings	参数解释： Redis实例进行添加副本时必选，指定每个副本所在的可用区Code，使用前需要先确认该可用区是否有足够资源。集群实例删除副本时必选。 具体查询方法，请参考 查询可用区信息 。
node_list	否	Array of strings	参数解释： Redis主备或读写分离实例进行删除副本时必选，指定需要删除的节点ID，目前仅支持一次删除一个副本。 节点ID查询方法，请参考 查询分片信息 。

参数	是否必选	参数类型	描述
execute_immediately	否	Boolean	参数解释： 是否立即变更。 约束限制： 不涉及。 取值范围： • true: 立即变更。 • false: 可维护时间窗内进行变更。目前只有部分场景支持在可维护时间窗内变更，请参考变更规格。 默认取值： true。
enable_ipv6	否	Boolean	参数解释： 是否开启IPv6。 约束限制： 不涉及。 取值范围： true：开启IPv6。 false：关闭IPv6。 默认取值： false。
spec_param	否	SpecParam object	参数解释： 线性规格定义参数。 约束限制： 不涉及。

表 5-58 BssParamEntity

参数	是否必选	参数类型	描述
is_auto_pay	否	String	参数解释： 功能说明：下单订购后，是否自动从客户的账户中支付；默认是“不自动支付”。 自动支付时，只能使用账户的现金支付；如果要使用代金券，请选择不自动支付，然后在用户费用中心，选择代金券支付。 如果包周期购买，没有设置成自动支付，即设置为false时，在变更规格或创建实例之后，用户必须在“费用中心 > 我的订单”，完成订单支付，否则订单一直在支付中，实例没有变更或者创建成功。 约束限制： 不涉及。 取值范围： • true: 是（自动支付，从账户余额自动扣费） • false: 否（默认值，只提交订单不支付，需要客户手动去支付） 默认取值： false

表 5-59 SpecParam

参数	是否必选	参数类型	描述
sharding_count	否	Integer	参数解释： 实例分片数。 约束限制： 不涉及。 取值范围： 1-128。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
replica_count	否	Integer	参数解释： 实例副本数。 约束限制： 不涉及。 取值范围： 1-10 默认取值： 不涉及。
cache_mode	否	String	参数解释： 缓存实例类型。 约束限制： 不涉及。 取值范围： • single：表示单机实例。 • ha：表示主备实例。 • cluster：表示cluster集群实例。 • proxy：表示Proxy集群实例。 • ha_rw_split：表示读写分离实例。 默认取值： 不涉及。

响应参数

状态码：204
缓存实例规格变更成功。
无

请求示例

- 变更实例规格，新的缓存实例规格为3副本1GB的主备实例。
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/resize

```
{
  "spec_code": "redis.ha.xu1.large.r3.1",
  "new_capacity": 1,
  "change_type": "createReplication",
  "available_zones": [ "e7afd64502d64fe3bfb60c2c82ec0ec6",
"d90ff6d692954373bf53be49cf3900cb" ],
  "execute_immediately": true
}
```

- 变更实例规格，新的缓存实例规格为2副本8GB的cluster集群实例。

```
{
  "spec_code": "redis.cluster.xu1.large.r2.8",
  "new_capacity": 8,
  "reserved_ip": [ "192.168.0.125" ],
  "change_type": "deleteReplication",
  "execute_immediately": true
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

- 变更实例规格，新的缓存实例规格为3副本1GB的主备实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ResizeInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        ResizeInstanceRequest request = new ResizeInstanceRequest();
        request.withInstanceId("{instance_id}");
        ResizeInstanceBody body = new ResizeInstanceBody();
        List<String> listbodyAvailableZones = new ArrayList<>();
        listbodyAvailableZones.add("e7afd64502d64fe3bfb60c2c82ec0ec6");
        listbodyAvailableZones.add("d90ff6d692954373bf53be49cf3900cb");
        body.withExecuteImmediately(true);
        body.withAvailableZones(listbodyAvailableZones);
        body.withChangeType(ResizeInstanceBody.ChangeTypeEnum.fromValue("createReplication"));
```

```
body.withNewCapacity(1);
body.withSpecCode("redis.ha.xu1.large.r3.1");
request.withBody(body);
try {
    ResizeInstanceResponse response = client.resizeInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

- 变更实例规格，新的缓存实例规格为2副本8GB的cluster集群实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ResizeInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeInstanceRequest request = new ResizeInstanceRequest();
        request.withInstanceId("{instance_id}");
        ResizeInstanceBody body = new ResizeInstanceBody();
        List<String> listbodyReservedIp = new ArrayList<>();
        listbodyReservedIp.add("192.168.0.125");
        body.withExecuteImmediately(true);
        body.withChangeType(ResizeInstanceBody.ChangeTypeEnum.fromValue("deleteReplication"));
        body.withReservedIp(listbodyReservedIp);
        body.withNewCapacity(8);
        body.withSpecCode("redis.cluster.xu1.large.r2.8");
        request.withBody(body);
        try {
            ResizeInstanceResponse response = client.resizeInstance(request);
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- 变更实例规格，新的缓存实例规格为3副本1GB的主备实例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeInstanceRequest()
        request.instance_id = "{instance_id}"
        listAvailableZonesbody = [
            "e7afd64502d64fe3bfb60c2c82ec0ec6",
            "d90ff6d692954373bf53be49cf3900cb"
        ]
        request.body = ResizeInstanceBody(
            execute_immediately=True,
            available_zones=listAvailableZonesbody,
            change_type="createReplication",
            new_capacity=1,
            spec_code="redis.ha.xu1.large.r3.1"
        )
        response = client.resize_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 变更实例规格，新的缓存实例规格为2副本8GB的cluster集群实例。

```
# coding: utf-8
```



```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeInstanceRequest()
        request.instance_id = "{instance_id}"
        listReservedIpbody = [
            "192.168.0.125"
        ]
        request.body = ResizeInstanceBody(
            execute_immediately=True,
            change_type="deleteReplication",
            reserved_ip=listReservedIpbody,
            new_capacity=8,
            spec_code="redis.cluster.xu1.large.r2.8"
        )
        response = client.resize_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 变更实例规格，新的缓存实例规格为3副本1GB的主备实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResizeInstanceRequest{}
request.InstanceId = "{instance_id}"
var listAvailableZonesbody = []string{
    "e7afd64502d64fe3bfb60c2c82ec0ec6",
    "d90ff6d692954373bf53be49cf3900cb",
}
executelmmediatelyResizeInstanceBody:= true
changeTypeResizeInstanceBody:=
model.GetResizeInstanceBodyChangeTypeEnum().CREATE_REPLICATION
request.Body = &model.ResizeInstanceBody{
    Executelmmediately: &executelmmediatelyResizeInstanceBody,
    AvailableZones: &listAvailableZonesbody,
    ChangeType: &changeTypeResizeInstanceBody,
    NewCapacity: int32(1),
    SpecCode: "redis.ha.xu1.large.r3.1",
}
response, err := client.ResizeInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 变更实例规格，新的缓存实例规格为2副本8GB的cluster集群实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ResizeInstanceRequest{}
request.InstanceId = "{instance_id}"
var listReservedIpbody = []string{
    "192.168.0.125",
}
executelmmediatelyResizeInstanceBody:= true
changeTypeResizeInstanceBody:=
model.GetResizeInstanceBodyChangeTypeEnum().DELETE_REPLICATION
request.Body = &model.ResizeInstanceBody{
    Executelmmediately: &executelmmediatelyResizeInstanceBody,
    ChangeType: &changeTypeResizeInstanceBody,
    ReservedIp: &listReservedIpbody,
    NewCapacity: int32(8),
    SpecCode: "redis.cluster.xu1.large.r2.8",
}
response, err := client.ResizeInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	缓存实例规格变更成功。

错误码

请参见[错误码](#)。

5.1.8 获取可删除副本

功能介绍

获取可删除的副本。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/deletable-replication

表 5-60 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

无

响应参数

状态码：200

表 5-61 响应 Body 参数

参数	参数类型	描述
check_result	Boolean	是否有可供选择的副本组进行删除。
available_zone	String	可选的可用区ID列表。
replication_list	Array of ReplicationInfo objects	可选的保留节点列表。

表 5-62 ReplicationInfo

参数	参数类型	描述
replication_id	String	副本ID。
node_id	String	节点ID。
replication_ip	String	副本的IP和端口。
group_id	String	分片ID。
group_name	String	分片名称。
available_zone	String	可用区ID。

状态码：400

表 5-63 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-64 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/deletable-replication
```

响应示例

状态码：200

获取可删除的副本成功。

```
{
  "check_result" : true,
  "available_zone" : "9f1c5806706d4c1fb0eb72f0a9b18c77",
  "replication_list" : [ {
    "replication_id" : "435a6322-0c64-4d0e-8bc2-8adefa2eca26",
    "node_id" : "8abf6e1e7fd3e695017fd4046bb9006b",
    "replication_ip" : "192.168.73.139:6379",
    "group_id" : "cbf8deba-d69f-41b8-8556-83337d32ad40",
    "group_name" : "group-2",
    "available_zone" : "9f1c5806706d4c1fb0eb72f0a9b18c77"
  }, {
    "replication_id" : "193683cf-f693-4a6a-950a-20f615ed6c47",
    "node_id" : "8abf6e1e7fd3e695017fd4046bb8006a",
    "replication_ip" : "192.168.67.86:6379",
    "group_id" : "7a153688-d31d-4541-b635-191885c71e93",
    "group_name" : "group-1",
    "available_zone" : "9f1c5806706d4c1fb0eb72f0a9b18c77"
  }, {
    "replication_id" : "ae38792f-60b7-4c8c-b1dc-175f2151f85f",
    "node_id" : "8abf6e1e7fd3e695017fd4046bb40069",
    "replication_ip" : "192.168.66.157:6379",
    "group_id" : "4f26a524-1525-4975-82fa-b2291091f9e7",
    "group_name" : "group-0",
    "available_zone" : "9f1c5806706d4c1fb0eb72f0a9b18c77"
  }
]
```

状态码：400

非法请求。

```
{
  "error_code" : 111400800,
  "error_msg" : "Only native cluster supported."
}
```

状态码：500

内部服务错误。

```
{
  "error_code" : 111500024,
  "error_msg" : "Query instance error."
}
```

状态码

状态码	描述
200	获取可删除的副本成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.1.9 包周期实例变更规格

功能介绍

包周期实例变更规格。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/orders/instances/{instance_id}/resize

表 5-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-66 请求 Body 参数

参数	是否必选	参数类型	描述
auto_open_security_group_rule	否	Boolean	自动开启安全组规则。
execute_immediately	否	Boolean	直接启动规格变更。
new_capacity	否	Float	新的容量，单位是GB。
password	否	String	密码。
spec_code	否	String	产品规格编码。

响应参数

状态码：200

表 5-67 响应 Body 参数

参数	参数类型	描述
order_id	String	订单ID，仅在创建包周期实例时返回。

状态码：400

表 5-68 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-69 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-70 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-71 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-72 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/orders/instances/{instance_id}/resize
{
  "auto_open_security_group_rule" : false,
  "execute_immediately" : true,
  "new_capacity" : 0.125,
  "password" : "",
  "spec_code" : "redis.single.xu1.tiny.128"
}
```

响应示例

状态码：200

请求变更成功。

```
{
  "order_id" : "CS2306141625TRHE4"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateResizeOrderSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateResizeOrderRequest request = new CreateResizeOrderRequest();
        request.withInstanceId("{instance_id}");
        CreateResizeOrderRequestBody body = new CreateResizeOrderRequestBody();
        body.withSpecCode("redis.single.xu1.tiny.128");
        body.withPassword("");
        body.withNewCapacity(0.125f);
        body.withExecuteImmediately(true);
        body.withAutoOpenSecurityGroupRule(false);
        request.withBody(body);
```

```
try {
    CreateResizeOrderResponse response = client.createResizeOrder(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateResizeOrderRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateResizeOrderRequestBody(
            spec_code="redis.single.xu1.tiny.128",
            password="",
            new_capacity=0.125,
            execute_immediately=True,
            auto_open_security_group_rule=False
        )
        response = client.create_resize_order(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
dcscfg := &model.CreateDcsConfigRequest{
    DcsConfig: &model.DcsConfig{
        Region: "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
    }
}

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateResizeOrderRequest{
        InstanceId: "{instance_id}"
        specCodeCreateResizeOrderRequestBody:= "redis.single.xu1.tiny.128"
        passwordCreateResizeOrderRequestBody:= ""
        newCapacityCreateResizeOrderRequestBody:= float32(0.125)
        executeImmediatelyCreateResizeOrderRequestBody:= true
        autoOpenSecurityGroupRuleCreateResizeOrderRequestBody:= false
        request.Body = &model.CreateResizeOrderRequestBody{
            SpecCode: &specCodeCreateResizeOrderRequestBody,
            Password: &passwordCreateResizeOrderRequestBody,
            NewCapacity: &newCapacityCreateResizeOrderRequestBody,
            ExecuteImmediately: &executeImmediatelyCreateResizeOrderRequestBody,
            AutoOpenSecurityGroupRule: &autoOpenSecurityGroupRuleCreateResizeOrderRequestBody,
        }
        response, err := client.CreateResizeOrder(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	请求变更成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。

状态码	描述
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2 实例管理

5.2.1 更新实例带宽弹性伸缩策略

功能介绍

更新实例带宽弹性伸缩策略。暂不支持实例带宽自动回缩。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth

表 5-73 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-74 请求 Body 参数

参数	是否必选	参数类型	描述
window_size	是	Integer	带宽弹性的观测窗口，单位：分钟。支持的取值：1、5、10、15、30。

参数	是否必选	参数类型	描述
bandwidth_usage_upper_threshold	是	Integer	触发带宽自动扩展的带宽平均使用率阈值，单位：百分比。支持的取值：50、60、70、80、90、95。
scale_out_cooldown	是	Integer	带宽扩展操作的静默时间（两次带宽扩展操作之间的最小间隔时间），单位：秒。 默认值：0。取值范围：0~86400。
scale_in_enabled	否	Boolean	是否启用带宽自动回缩。默认值：false。该参数暂未启用。
bandwidth_usage_lower_threshold	否	Integer	触发带宽自动回缩的带宽平均使用率阈值，单位：百分比。支持的取值：10、20、30。该参数暂未启用。
scale_in_cooldown	否	Integer	带宽回缩操作的静默时间（两次带宽回缩操作之间的最小间隔时间），单位：秒。该参数暂未启用。 默认值：300。取值范围：0~86400。

响应参数

状态码：200

表 5-75 响应 Body 参数

参数	参数类型	描述
window_size	Integer	带宽弹性的观测窗口，单位：分钟。支持的取值：1、5、10、15、30。
bandwidth_usage_upper_threshold	Integer	触发带宽自动扩展的带宽平均使用率阈值，单位：百分比。支持的取值：50、60、70、80、90、95。
scale_out_cooldown	Integer	带宽扩展操作的静默时间（两次带宽扩展操作之间的最小间隔时间），单位：秒。 默认值：0。取值范围：0~86400。
scale_in_enabled	Boolean	是否启用带宽自动回缩。默认值：false。该参数暂未启用。

参数	参数类型	描述
bandwidth_usage_lower_threshold	Integer	触发带宽自动回缩的带宽平均使用率阈值，单位：百分比。支持的取值：10、20、30。该参数暂未启用。
scale_in_cooldown	Integer	带宽回缩操作的静默时间（两次带宽回缩操作之间的最小间隔时间），单位：秒。该参数暂未启用。 默认值：300。取值范围：0~86400。

状态码：400

表 5-76 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-77 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-78 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-79 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-80 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth
{
```

```
"window_size" : 5,  
"bandwidth_usage_upper_threshold" : 70,  
"scale_out_cooldown" : 0,  
"scale_in_enabled" : true,  
"bandwidth_usage_lower_threshold" : 30,  
"scale_in_cooldown" : 300  
}
```

响应示例

状态码：200

更新实例弹性伸缩策略响应体。

```
{  
  "window_size" : 5,  
  "bandwidth_usage_upper_threshold" : 70,  
  "scale_out_cooldown" : 0,  
  "scale_in_enabled" : true,  
  "bandwidth_usage_lower_threshold" : 30,  
  "scale_in_cooldown" : 300  
}
```

状态码：400

非法请求。

```
{  
  "error_code" : "DCS.4063",  
  "error_msg" : "Invalid window_size parameter in the request."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
public class UpdateInstanceBandwidthAutoScalingPolicySolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateInstanceBandwidthAutoScalingPolicyRequest request = new
UpdateInstanceBandwidthAutoScalingPolicyRequest();
request.withInstanceId("{instance_id}");
UpdateInstanceBandwidthAutoScalingPolicyRequestBody body = new
UpdateInstanceBandwidthAutoScalingPolicyRequestBody();
body.withScaleInCooldown(300);
body.withScaleOutCooldown(0);
body.withBandwidthUsageLowerThreshold(30);
body.withScaleInEnabled(true);

body.withBandwidthUsageUpperThreshold(UpdateInstanceBandwidthAutoScalingPolicyRequestBody.BandwidthUsageUpperThresholdEnum.NUMBER_70);

body.withWindowSize(UpdateInstanceBandwidthAutoScalingPolicyRequestBody.WindowSizeEnum.NUMBER_5);
request.withBody(body);
try {
    UpdateInstanceBandwidthAutoScalingPolicyResponse response =
client.updateInstanceBandwidthAutoScalingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceBandwidthAutoScalingPolicyRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateInstanceBandwidthAutoScalingPolicyRequestBody()
```

```
        scale_in_cooldown=300,  
        scale_out_cooldown=0,  
        bandwidth_usage_lower_threshold=30,  
        scale_in_enabled=True,  
        bandwidth_usage_upper_threshold=70,  
        window_size=5  
    )  
    response = client.update_instance_bandwidth_auto_scaling_policy(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dcs.NewDcsClient(  
        dcs.DcsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateInstanceBandwidthAutoScalingPolicyRequest{  
        request.InstanceId = "{instance_id}"  
        scaleInCooldownUpdateInstanceBandwidthAutoScalingPolicyRequestBody := int32(300)  
        bandwidthUsageLowerThresholdUpdateInstanceBandwidthAutoScalingPolicyRequestBody := int32(30)  
        scaleInEnabledUpdateInstanceBandwidthAutoScalingPolicyRequestBody := true  
        request.Body = &model.UpdateInstanceBandwidthAutoScalingPolicyRequestBody{  
            ScaleInCooldown: &scaleInCooldownUpdateInstanceBandwidthAutoScalingPolicyRequestBody,  
            ScaleOutCooldown: int32(0),  
            BandwidthUsageLowerThreshold:  
&bandwidthUsageLowerThresholdUpdateInstanceBandwidthAutoScalingPolicyRequestBody,  
            ScaleInEnabled: &scaleInEnabledUpdateInstanceBandwidthAutoScalingPolicyRequestBody,  
            BandwidthUsageUpperThreshold:  
model.GetUpdateInstanceBandwidthAutoScalingPolicyRequestBodyBandwidthUsageUpperThresholdEnum().  
E_70,  
            WindowSize:  
model.GetUpdateInstanceBandwidthAutoScalingPolicyRequestBodyWindowSizeEnum().E_5,  
        }  
        response, err := client.UpdateInstanceBandwidthAutoScalingPolicy(request)  
        if err == nil {  
            fmt.Printf("%v\n", response)
```

```
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新实例弹性伸缩策略响应体。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.2 查询实例带宽弹性伸缩策略

功能介绍

查询实例带宽弹性伸缩策略。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth

表 5-81 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200

表 5-82 响应 Body 参数

参数	参数类型	描述
window_size	Integer	带宽弹性的观测窗口，单位：分钟。支持的取值：1、5、10、15、30。
bandwidth_usage_upper_threshold	Integer	触发带宽自动扩展的带宽平均使用率阈值，单位：百分比。支持的取值：50、60、70、80、90、95。
scale_out_cooldown	Integer	带宽扩展操作的静默时间（两次带宽扩展操作之间的最小间隔时间），单位：秒。 默认值：0。取值范围：0~86400。
scale_in_enabled	Boolean	是否启用带宽自动回缩。默认值：false。该参数暂未启用。
bandwidth_usage_lower_threshold	Integer	触发带宽自动回缩的带宽平均使用率阈值，单位：百分比。支持的取值：10、20、30。该参数暂未启用。
scale_in_cooldown	Integer	带宽回缩操作的静默时间（两次带宽回缩操作之间的最小间隔时间），单位：秒。该参数暂未启用。 默认值：300。取值范围：0~86400。

状态码：400

表 5-83 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-84 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-85 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-86 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-87 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth
```

响应示例

状态码：200

查询实例带宽弹性伸缩策略响应体。

```
{
  "window_size" : 5,
  "bandwidth_usage_upper_threshold" : 70,
  "scale_out_cooldown" : 0,
  "scale_in_enabled" : true,
  "bandwidth_usage_lower_threshold" : 30,
  "scale_in_cooldown" : 300
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4062",
  "error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowInstanceBandwidthAutoScalingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();

        ShowInstanceBandwidthAutoScalingPolicyRequest request = new
        ShowInstanceBandwidthAutoScalingPolicyRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowInstanceBandwidthAutoScalingPolicyResponse response =
            client.showInstanceBandwidthAutoScalingPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
```

```
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowInstanceBandwidthAutoScalingPolicyRequest()
    request.instance_id = "{instance_id}"
    response = client.show_instance_bandwidth_auto_scaling_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceBandwidthAutoScalingPolicyRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceBandwidthAutoScalingPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例带宽弹性伸缩策略响应体。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.3 删除实例带宽弹性伸缩策略

功能介绍

删除实例带宽弹性伸缩策略。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth

表 5-88 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200
删除实例带宽弹性伸缩策略成功。

状态码：400

表 5-89 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-90 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-91 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-92 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-93 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/autoscaling-policy/bandwidth
```

响应示例

状态码：400

非法请求。

```
{  
  "error_code" : "DCS.4062",  
}
```



```
"error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteInstanceBandwidthAutoScalingPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        DeleteInstanceBandwidthAutoScalingPolicyRequest request = new
        DeleteInstanceBandwidthAutoScalingPolicyRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteInstanceBandwidthAutoScalingPolicyResponse response =
            client.deleteInstanceBandwidthAutoScalingPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteInstanceBandwidthAutoScalingPolicyRequest()
        request.instance_id = "{instance_id}"
        response = client.delete_instance_bandwidth_auto_scaling_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteInstanceBandwidthAutoScalingPolicyRequest{}
```

```
request.InstanceId = "{instance_id}"
response, err := client.DeleteInstanceBandwidthAutoScalingPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除实例带宽弹性伸缩策略成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.4 查询实例参数修改记录详情

功能介绍

查询实例参数修改记录详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/config-histories/{history_id}

表 5-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
history_id	是	String	实例参数修改记录ID，需要调用 查询实例参数修改记录列表 接口获取。

请求参数

无

响应参数

状态码：200
查询实例参数修改记录详情成功。

状态码：400

表 5-95 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-96 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/config-histories/{history_id}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4987",
  "error_msg" : "Query config update history data failed."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowConfigHistoryDetailSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowConfigHistoryDetailRequest request = new ShowConfigHistoryDetailRequest();
request.withInstanceId("{instance_id}");
request.withHistoryId("{history_id}");
try {
    ShowConfigHistoryDetailResponse response = client.showConfigHistoryDetail(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConfigHistoryDetailRequest()
        request.instance_id = "{instance_id}"
        request.history_id = "{history_id}"
```

```
response = client.show_config_history_detail(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConfigHistoryDetailRequest{}
    request.InstanceId = "{instance_id}"
    request.HistoryId = "{history_id}"
    response, err := client.ShowConfigHistoryDetail(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例参数修改记录详情成功。
400	非法请求。

状态码	描述
500	内部服务器错误。

错误码

请参见[错误码](#)。

5.2.5 主备切换

功能介绍

切换实例主备节点，只有主备实例支持该操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/swap

表 5-97 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

无

响应参数

状态码：204
主备切换成功。

状态码：400

表 5-98 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-99 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-100 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-101 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-102 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/swap
```

响应示例

状态码：400

非法请求。

```
{
  "error_msg": "Master standby swap is not supported.",
  "error_code": "DCS.4855",
  "error_ext_msg": null
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ChangeMasterStandbySolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DcsClient client = DcsClient.newBuilder()
        .withCredential(auth)
        .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
        .build();
    ChangeMasterStandbyRequest request = new ChangeMasterStandbyRequest();
    request.withInstanceId("{instance_id}");
    try {
        ChangeMasterStandbyResponse response = client.changeMasterStandby(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeMasterStandbyRequest()
        request.instance_id = "{instance_id}"
```

```
response = client.change_master_standby(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeMasterStandbyRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ChangeMasterStandby(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	主备切换成功。
400	非法请求。

状态码	描述
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.6 修改密码

功能介绍

修改缓存实例的密码。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/password

表 5-103 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-104 请求 Body 参数

参数	是否必选	参数类型	描述
old_password	是	String	参数解释： 实例旧密码。 约束限制： 密码长度在8到64位之间。 取值范围： 至少包含如下四种字符中的三种： • 小写字母 • 大写字母 • 数字 • 特殊字符包括 (`~!@#\$%^&*()-_+=+ {}<.>/?) 默认取值： 不涉及。

参数	是否必选	参数类型	描述
new_password	是	String	参数解释： 实例新密码。 约束限制： 新密码不能与旧密码相同，密码长度在8到64位之间。 取值范围： 至少包含如下四种字符中的三种： • 小写字母 • 大写字母 • 数字 • 特殊字符包括（`~!@#\$%^&*()-_+=+ {>,.<./?`） 默认取值： 不涉及。

响应参数

状态码：200

表 5-105 响应 Body 参数

参数	参数类型	描述
lock_time	String	参数解释： 锁定时间（单位：分钟）。如果重复5次输入错误的旧密码，实例会被锁定，锁定期间不允许修改或重置密码。 取值范围： 0-5
result	String	参数解释： 密码修改结果。 取值范围： • success：密码修改成功。 • passwordFailed：密码验证失败。 • locked：已锁定。 • failed：密码修改失败。

参数	参数类型	描述
lock_time_left	String	参数解释： 锁定剩余时间（单位：分钟）。 取值范围： 0-5
retry_times_left	String	参数解释： 密码验证剩余次数。 取值范围： 不涉及。
message	String	参数解释： 修改结果说明信息。 取值范围： 不涉及。

状态码：400

表 5-106 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-107 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

修改缓存实例密码，配置旧密码和新密码。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/password
{
  "old_password": "XXXXXX",
  "new_password": "XXXXXX"
}
```

响应示例

状态码：200

参数解释：

修改密码成功。

```
{
  "lock_time": "0",
  "result": "success",
  "lock_time_left": "0",
  "retry_times_left": "5",
  "message": "Modify DCSInstance password success."
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改缓存实例密码，配置旧密码和新密码。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdatePasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdatePasswordRequest request = new UpdatePasswordRequest();
        request.withInstanceId("{instance_id}");
        ModifyInstancePasswordBody body = new ModifyInstancePasswordBody();
        body.withNewPassword("XXXXXX");
        body.withOldPassword("XXXXXX");
        request.withBody(body);
        try {
            UpdatePasswordResponse response = client.updatePassword(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改缓存实例密码，配置旧密码和新密码。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdatePasswordRequest()
    request.instance_id = "{instance_id}"
    request.body = ModifyInstancePasswordBody(
        new_password="XXXXXX",
        old_password="XXXXXX"
    )
    response = client.update_password(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

修改缓存实例密码，配置旧密码和新密码。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePasswordRequest{}
    request.InstanceId = "{instance_id}"
```

```
request.Body = &model.ModifyInstancePasswordBody{
    NewPassword: "XXXXXX",
    OldPassword: "XXXXXX",
}
response, err := client.UpdatePassword(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 修改密码成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.7 重置密码

功能介绍

重置缓存实例的密码。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/password/reset

表 5-108 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-109 请求 Body 参数

参数	是否必选	参数类型	描述
new_password	否	String	参数解释： 重置的新密码。当“no_password_access”配置为“false”或未配置时，请求消息中须包含password参数。 约束限制： 密码长度在8到64位之间。 取值范围： 至少包含如下四种字符中的三种： • 小写字母 • 大写字母 • 数字 • 特殊字符包括（`~!@#\$%^&*()-_+= {>,.<./?`） 默认取值： 不涉及。
no_password_access	否	Boolean	参数解释： 是否重置为免密码访问缓存实例。 约束限制： 不涉及。 取值范围： • true：重置为免密访问的缓存实例。 • false：重置为密码访问的缓存实例。 默认取值： false

响应参数

状态码：200

表 5-110 响应 Body 参数

参数	参数类型	描述
retry_times_left	String	参数解释： 密码验证剩余次数。重置密码不涉及密码验证，该参数已废弃，默认返回5。 取值范围： 不涉及。
lock_time	String	参数解释： 锁定时间（单位：分钟）。重置密码不涉及锁定，该参数已废弃，默认返回0。 取值范围： 不涉及。
lock_time_left	String	参数解释： 锁定剩余时间（单位：分钟）。重置密码不涉及锁定，该参数已废弃，默认返回0。 取值范围： 不涉及。
code	String	参数解释： 重置密码结果编号。 取值范围： 1：重置密码成功。
message	String	参数解释： 重置密码结果信息。 取值范围： success
ext_message	String	参数解释： 重置密码错误信息，若重置密码成功，则为null。 取值范围： 不涉及。

状态码：400

表 5-111 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-112 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

重置缓存实例密码，配置新密码。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/password/reset

{
  "new_password": "xxxxxx",
  "no_password_access": false
}
```

响应示例

状态码：200

参数解释：

重置缓存实例的密码成功。

```
{
  "lock_time": "0",
  "lock_time_left": "0",
  "retry_times_left": "5",
  "code": "1",
  "message": "success",
  "ext_message": null
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4839",
  "error_msg": "is not support reset instance password."
}
```

SDK 代码示例

SDK代码示例如下。

Java

重置缓存实例密码，配置新密码。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ResetPasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ResetPasswordRequest request = new ResetPasswordRequest();
request.withInstanceId("{instance_id}");
ResetInstancePasswordBody body = new ResetInstancePasswordBody();
body.withNoPasswordAccess(false);
body.withNewPassword("xxxxxx");
request.withBody(body);
try {
    ResetPasswordResponse response = client.resetPassword(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

重置缓存实例密码，配置新密码。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetPasswordRequest()
        request.instance_id = "{instance_id}"
        request.body = ResetInstancePasswordBody(
            no_password_access=False,
            new_password="xxxxxx"
        )
        response = client.reset_password(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

重置缓存实例密码，配置新密码。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetPasswordRequest{}
    request.InstanceId = "{instance_id}"
    noPasswordAccessResetInstancePasswordBody := false
    newPasswordResetInstancePasswordBody := "xxxxxx"
    request.Body = &model.ResetInstancePasswordBody{
        NoPasswordAccess: &noPasswordAccessResetInstancePasswordBody,
        NewPassword: &newPasswordResetInstancePasswordBody,
    }
    response, err := client.ResetPassword(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 重置缓存实例的密码成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.8 变更可用区

功能介绍

迁移缓存实例可用区，完成单可用区实例跨可用区改造。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/available-zones

表 5-113 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-114 请求 Body 参数

参数	是否必选	参数类型	描述
new_available_zones	否	Array of strings	参数解释： 新可用区。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
execute_immediately	否	Boolean	参数解释： 是否立即执行变更。 约束限制： 不涉及。 取值范围： true：立即执行变更。 false：暂不执行变更。 默认取值： true

响应参数

状态码：204
变更可用区请求提交成功。

状态码：400

表 5-115 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-116 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

变更单可用区实例为多可用区。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/available-zones

{
  "new_available_zones": [ "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef", "2eaaff66-fbb0-47ad-b6cb-fb87f5b12fbe" ],
  "execute_immediately": true
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

变更单可用区实例为多可用区。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class MigrateAzSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        MigrateAzRequest request = new MigrateAzRequest();
        request.withInstanceId("{instance_id}");
        MigrateAZRequestBody body = new MigrateAZRequestBody();
        List<String> listbodyNewAvailableZones = new ArrayList<>();
        listbodyNewAvailableZones.add("2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef");
        listbodyNewAvailableZones.add("2eaaff66-fbb0-47ad-b6cb-fb87f5b12fbe");
        body.withExecuteImmediately(true);
        body.withNewAvailableZones(listbodyNewAvailableZones);
        request.withBody(body);
        try {
            MigrateAzResponse response = client.migrateAz(request);
        }
    }
}
```



```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

变更单可用区实例为多可用区。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = MigrateAzRequest()
        request.instance_id = "{instance_id}"
        listNewAvailableZonesbody = [
            "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",
            "2eaaff66-fbb0-47ad-b6cb-fb87f5b12fbe"
        ]
        request.body = MigrateAZRequestBody(
            execute_immediately=True,
            new_available_zones=listNewAvailableZonesbody
        )
        response = client.migrate_az(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

变更单可用区实例为多可用区。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.MigrateAzRequest{}
    request.InstanceId = "{instance_id}"
    var listNewAvailableZonesbody = []string{
        "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",
        "2eaaff66-fbb0-47ad-b6cb-fb87f5b12fbe",
    }
    executelImmediatelyMigrateAzRequestBody:= true
    request.Body = &model.MigrateAzRequestBody{
        ExecutelImmediately: &executelImmediatelyMigrateAzRequestBody,
        NewAvailableZones: &listNewAvailableZonesbody,
    }
    response, err := client.MigrateAz(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	变更可用区请求提交成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.9 查询实例状态

功能介绍

查询该租户在当前区域下不同状态的实例数。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/status

表 5-117 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-118 Query 参数

参数	是否必选	参数类型	描述
include_failure	否	String	参数解释： 是否返回创建失败的实例数。 约束限制： 不涉及。 取值范围： • true：返回的统计包括创建失败的实例数。 • false：返回的统计不包括创建失败的实例数。 默认取值： false

请求参数

无

响应参数

状态码：200

表 5-119 响应 Body 参数

参数	参数类型	描述
redis	StatusStatistic object	参数解释： Redis实例数量统计信息。 取值范围： 不涉及。
memcached	StatusStatistic object	参数解释： Memcached实例数量统计信息。 取值范围： 不涉及。
paying_count	Integer	参数解释： 支付中的实例数。 取值范围： 不涉及。

参数	参数类型	描述
freezing_count	Integer	参数解释： 冻结中的实例数。 取值范围： 不涉及。
migrating_count	Integer	参数解释： 迁移中的实例数。 取值范围： 不涉及。
flushing_count	Integer	参数解释： 清空中的实例数。 取值范围： 不涉及。
upgrading_count	Integer	参数解释： 升级中的实例数。 取值范围： 不涉及。
restoring_count	Integer	参数解释： 恢复中的实例数。 取值范围： 不涉及。
extending_count	Integer	参数解释： 扩容中的实例数。 取值范围： 不涉及。
creating_count	Integer	参数解释： 正在创建的实例数。 取值范围： 不涉及。
running_count	Integer	参数解释： 正在运行的实例数。 取值范围： 不涉及。
error_count	Integer	参数解释： 异常的实例数。 取值范围： 不涉及。

参数	参数类型	描述
frozen_count	Integer	参数解释： 已冻结的实例数。 取值范围： 不涉及。
createfailed_count	Integer	参数解释： 创建失败的实例数。 取值范围： 不涉及。
restarting_count	Integer	参数解释： 正在重启的实例数。 取值范围： 不涉及。

表 5-120 StatusStatistic

参数	参数类型	描述
paying_count	Integer	参数解释： 支付中的实例数。 取值范围： 不涉及。
freezing_count	Integer	参数解释： 冻结中的实例数。 取值范围： 不涉及。
migrating_count	Integer	参数解释： 迁移中的实例数。 取值范围： 不涉及。
flushing_count	Integer	参数解释： 清空中的实例数。 取值范围： 不涉及。
upgrading_count	Integer	参数解释： 升级中的实例数。 取值范围： 不涉及。

参数	参数类型	描述
restoring_count	Integer	参数解释： 恢复中的实例数。 取值范围： 不涉及。
extending_count	Integer	参数解释： 扩容中的实例数。 取值范围： 不涉及。
creating_count	Integer	参数解释： 正在创建的实例数。 取值范围： 不涉及。
running_count	Integer	参数解释： 正在运行的实例数。 取值范围： 不涉及。
error_count	Integer	参数解释： 异常的实例数。 取值范围： 不涉及。
frozen_count	Integer	参数解释： 已冻结的实例数。 取值范围： 不涉及。
createfailed_count	Integer	参数解释： 创建失败的实例数 取值范围： 不涉及。
restarting_count	Integer	参数解释： 正在重启的实例数 取值范围： 不涉及。

状态码：400

表 5-121 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-122 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/status?include_failure=true
```


响应示例

状态码：200

查询实例数量统计信息成功。

```
{
  "extending_count" : 0,
  "creating_count" : 0,
  "running_count" : 16,
  "error_count" : 0,
  "frozen_count" : 0,
  "createfailed_count" : 44,
  "restarting_count" : 0,
  "paying_count" : 0,
  "freezing_count" : 0,
  "migrating_count" : 0,
  "flushing_count" : 0,
  "upgrading_count" : 0,
  "redis" : {
    "extending_count" : 0,
    "creating_count" : 0,
    "running_count" : 8,
    "error_count" : 0,
    "frozen_count" : 0,
    "createfailed_count" : 22,
    "restarting_count" : 0,
    "paying_count" : 0,
    "freezing_count" : 0,
    "migrating_count" : 0,
    "flushing_count" : 0,
    "upgrading_count" : 0
  },
  "memcached" : {
    "extending_count" : 0,
    "creating_count" : 0,
    "running_count" : 8,
    "error_count" : 0,
    "frozen_count" : 0,
    "createfailed_count" : 22,
    "restarting_count" : 0,
    "paying_count" : 0,
    "freezing_count" : 0,
    "migrating_count" : 0,
    "flushing_count" : 0,
    "upgrading_count" : 0
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class ListNumberOfInstancesInDifferentStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListNumberOfInstancesInDifferentStatusRequest request = new
        ListNumberOfInstancesInDifferentStatusRequest();
        try {
            ListNumberOfInstancesInDifferentStatusResponse response =
            client.listNumberOfInstancesInDifferentStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ListNumberOfInstancesInDifferentStatusRequest()
response = client.list_number_of_instances_in_different_status(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNumberOfInstancesInDifferentStatusRequest{}
    response, err := client.ListNumberOfInstancesInDifferentStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例数量统计信息成功。
400	非法请求。

状态码	描述
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.10 重启实例或清空数据

功能介绍

重启运行中的DCS缓存实例。

清空Redis4.0及以上版本的实例数据，数据清空后，无法撤销，且无法恢复，请谨慎操作。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/status

表 5-123 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-124 请求 Body 参数

参数	是否必选	参数类型	描述
instances	否	Array of strings	参数解释： 实例的ID列表。
action	否	String	参数解释： 对实例的操作。 约束限制： 当前版本，只有Redis 4.0及以上实例支持清空数据功能，即flush操作。 取值范围： • restart：强制重启 • soft_restart：软重启，只重启进程 • flush：清空数据 默认取值： 不涉及。

响应参数

状态码：200

表 5-125 响应 Body 参数

参数	参数类型	描述
results	Array of BatchOpsResult objects	参数解释： 删除/重启/清空实例的结果。

表 5-126 BatchOpsResult

参数	参数类型	描述
result	String	参数解释： 操作结果。 取值范围： success: 删除/重启/清空实例成功。 failed: 删除/重启/清空实例失败。

参数	参数类型	描述
instance	String	参数解释： 缓存实例ID。 取值范围： 不涉及。

状态码：400

表 5-127 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-128 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

请求示例

强制重启缓存实例。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/status
{
  "instances": [ "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef" ],
  "action": "restart"
}
```

响应示例

状态码：200

参数解释：

重启实例或清除实例数据成功。

```
{
  "results": [ {
    "instance": "e3a7019c-8824-4c1a-8289-5300f19b9f64",
    "result": "success"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

强制重启缓存实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class RestartOrFlushInstancesSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
RestartOrFlushInstancesRequest request = new RestartOrFlushInstancesRequest();
ChangeInstanceStatusBody body = new ChangeInstanceStatusBody();
List<String> listbodyInstances = new ArrayList<>();
listbodyInstances.add("2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef");
body.withAction("restart");
body.withInstances(listbodyInstances);
request.withBody(body);
try {
    RestartOrFlushInstancesResponse response = client.restartOrFlushInstances(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

强制重启缓存实例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
```



```
.build()

try:
    request = RestartOrFlushInstancesRequest()
    listInstancesbody = [
        "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef"
    ]
    request.body = ChangeInstanceStatusBody(
        action="restart",
        instances=listInstancesbody
    )
    response = client.restart_or_flush_instances(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

强制重启缓存实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartOrFlushInstancesRequest{}
    var listInstancesbody = []string{
        "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",
    }
    actionChangeInstanceStatusBody := "restart"
    request.Body = &model.ChangeInstanceStatusBody{
        Action: &actionChangeInstanceStatusBody,
        Instances: &listInstancesbody,
    }
    response, err := client.RestartOrFlushInstances(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 重启实例或清除实例数据成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.11 查询运行中实例的统计信息

功能介绍

查询当前租户下处于“运行中”状态的缓存实例的统计信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/statistic

表 5-129 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-130 响应 Body 参数

参数	参数类型	描述
statistics	Array of InstanceStatistic objects	参数解释： 该租户下处于“运行中”状态的实例的统计信息。 取值范围： 不涉及。

表 5-131 InstanceStatistic

参数	参数类型	描述
input_kbps	String	参数解释： 缓存实例网络入流量，单位：kbit/s。 取值范围： 不涉及。
output_kbps	String	参数解释： 缓存实例网络出流量，单位：kbit/s。 取值范围： 不涉及。

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
keys	Long	参数解释： 缓存存储的数据条数。 取值范围： 不涉及。
used_memory	Long	参数解释： 缓存已经使用内存，单位：MB。 取值范围： 不涉及。
max_memory	Long	参数解释： 缓存的总内存，单位：MB。 取值范围： 不涉及。
cmd_get_count	Long	参数解释： 缓存get命令被调用次数。 取值范围： 不涉及。
cmd_set_count	Long	参数解释： 缓存set命令被调用次数。 取值范围： 不涉及。
used_cpu	String	参数解释： 缓存在用户态和核心态消耗CPU时间的 累计值，单位：秒。 取值范围： 不涉及。

状态码： 400

表 5-132 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-133 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/statistic
```

响应示例

状态码：200

查询实例统计信息成功。

```
{
  "statistics": [ {
    "output_kbps": "0.0",
    "instance_id": "e008652d-18e0-43ff-924e-072261e0372a",
    "keys": 0,
    "used_memory": 0,
    "cmd_get_count": 0,
    "used_cpu": "0.0",
    "cmd_set_count": 0,
    "input_kbps": "0.0",
    "max_memory": 460
  }, {
    "output_kbps": "0.0",
    "instance_id": "c577a1eb-33b7-42c7-8231-ad32358599ac",
    "keys": 0,
    "used_memory": 0,
    "cmd_get_count": 0,
    "used_cpu": "0.0",
    "cmd_set_count": 0,
    "input_kbps": "0.0",
    "max_memory": 460
  }, {
    "output_kbps": "1.19",
    "instance_id": "e8b98471-55d5-4695-b0bb-8f336a98e207",
    "keys": 0,
    "used_memory": 0,
    "cmd_get_count": 0,
    "used_cpu": "0.0",
    "cmd_set_count": 0,
    "input_kbps": "0.03",
    "max_memory": 460
  }, {
    "output_kbps": "0.0",
    "instance_id": "bc61c690-4b34-4cbe-9ce3-11246aea7aba",
    "keys": 0,
    "used_memory": 0,
    "cmd_get_count": 0,
    "used_cpu": "0.0",
    "cmd_set_count": 0,
    "input_kbps": "0.0",
    "max_memory": 6963
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class ListStatisticsOfRunningInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListStatisticsOfRunningInstancesRequest request = new ListStatisticsOfRunningInstancesRequest();
        try {
            ListStatisticsOfRunningInstancesResponse response =
client.listStatisticsOfRunningInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = ListStatisticsOfRunningInstancesRequest()
response = client.list_statistics_of_running_instances(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListStatisticsOfRunningInstancesRequest{}
    response, err := client.ListStatisticsOfRunningInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例统计信息成功。
400	非法请求。

状态码	描述
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.12 开启或关闭客户端 IP 透传

功能介绍

开启或关闭客户端IP透传。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/{instance_id}/client-ip-transparent-transmission

表 5-134 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-135 请求 Body 参数

参数	是否必选	参数类型	描述
transparent_client_ip_enable	否	Boolean	开启或关闭客户IP透传标志。

响应参数

状态码：200

表 5-136 响应 Body 参数

参数	参数类型	描述
job_id	String	开启或关闭客户IP透传的任务ID。

状态码：400

表 5-137 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-138 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/{instance_id}/client-ip-transparent-transmission
{
  "transparent_client_ip_enable" : true
}
```

响应示例

状态码：200

开启或关闭客户IP透传成功。

```
{
  "job_id" : "8a2c26c17fc0d77c017fc0dc647b0002"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4800",
  "error_msg" : "Invalid transparent_client_ip_enable in the request."
}
```

状态码：500

内部服务错误。

```
{
  "error_code" : "DCS.5000",
  "error_msg" : "Internal service error."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class UpdateClientIpTransparentTransmissionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateClientIpTransparentTransmissionRequest request = new
        UpdateClientIpTransparentTransmissionRequest();
        request.withInstanceId("{instance_id}");
        UpdateClientIpTransparentTransmissionRequestBody body = new
        UpdateClientIpTransparentTransmissionRequestBody();
        body.withTransparentClientIpEnable(true);
        request.withBody(body);
        try {
            UpdateClientIpTransparentTransmissionResponse response =
            client.updateClientIpTransparentTransmission(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateClientIpTransparentTransmissionRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdateClientIpTransparentTransmissionRequestBody(
        transparent_client_ip_enable=True
    )
    response = client.update_client_ip_transparent_transmission(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClientIpTransparentTransmissionRequest{
        request.InstanceId = "{instance_id}"
        transparentClientIpEnableUpdateClientIpTransparentTransmissionRequestBody:= true
        request.Body = &model.UpdateClientIpTransparentTransmissionRequestBody{
            TransparentClientIpEnable:
            &transparentClientIpEnableUpdateClientIpTransparentTransmissionRequestBody,
        }
        response, err := client.UpdateClientIpTransparentTransmission(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	开启或关闭客户IP透传成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.13 变更指定实例的带宽

功能介绍

Redis 4.0及之后版本的实例，支持临时扩容缓存实例带宽，用于临时解决业务流量高峰，带宽超限的问题。实例带宽扩容后，带宽为原带宽的2倍。Proxy集群实例带宽上限为10000Mbit/s，达到上限后无法扩容。一次带宽临时扩容的有效期为7天，到期后恢复为原带宽，失效1天后可再次申请扩容，一个实例最多扩容3次。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/bandwidth

表 5-139 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200
带宽变更成功。

状态码：400

表 5-140 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-141 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/bandwidth
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateInstanceBandwidthSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceBandwidthRequest request = new UpdateInstanceBandwidthRequest();
        request.withInstanceId("{instance_id}");
        try {
            UpdateInstanceBandwidthResponse response = client.updateInstanceBandwidth(request);
            System.out.println(response.toString());
        } catch (Exception e) {
            // Handle exceptions
        }
    }
}
```



```
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceBandwidthRequest()
        request.instance_id = "{instance_id}"
        response = client.update_instance_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateInstanceBandwidthRequest{
    request.InstanceId = "{instance_id}"
}
response, err := client.UpdateInstanceBandwidth(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	带宽变更成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.14 开启/修改实例公网访问

功能介绍

开启/修改实例公网访问。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/public-ip

表 5-142 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-143 请求 Body 参数

参数	是否必选	参数类型	描述
publicip_id	否	String	公网IP的ID，当Redis版本为3.0时，该参数为必填参数。
enable_ssl	否	Boolean	是否开启SSL，仅在开启SSL时有值，当Redis版本为3.0时，该参数为必填参数。
elb_id	否	String	公网访问绑定的ELB ID，当Redis为4.0及以上版本时，该参数为必填参数。

响应参数

状态码：200

表 5-144 响应 Body 参数

参数	参数类型	描述
job_id	String	请求提交任务ID。

状态码：204

开启/修改 Redis 3.0 实例公网访问成功。

状态码：400

表 5-145 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-146 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-147 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-148 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-149 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- 开启Redis 4.0及以上版本公网访问。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/public-ip
{
  "elb_id" : "3ee8da68-5a23-4ca4-b5e0-74ff996e16be"
}
```
- 开启Redis 3.0版本公网访问。

```
{
  "enable_ssl" : true,
  "publicip_id" : "00130845-97c3-43fa-88eb-85cfb08d87ca"
}
```

响应示例

状态码：200

提交开启实例公网访问任务成功。

```
{
  "job_id" : "ff808081816ba43101816ba4e0ea0000"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 开启Redis 4.0及以上版本公网访问。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdatePublicIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdatePublicIpRequest request = new UpdatePublicIpRequest();
        request.withInstanceId("{instance_id}");
        UpdatePublicIpRequestBody body = new UpdatePublicIpRequestBody();
        body.withElbId("3ee8da68-5a23-4ca4-b5e0-74ff996e16be");
        request.withBody(body);
        try {
            UpdatePublicIpResponse response = client.updatePublicIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

- 开启Redis 3.0版本公网访问。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdatePublicIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdatePublicIpRequest request = new UpdatePublicIpRequest();
        request.withInstanceId("{instance_id}");
        UpdatePublicIpRequestBody body = new UpdatePublicIpRequestBody();
        body.withEnableSsl(true);
        body.withPublicIpId("00130845-97c3-43fa-88eb-85cfb08d87ca");
        request.withBody(body);
        try {
            UpdatePublicIpResponse response = client.updatePublicIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 开启Redis 4.0及以上版本公网访问。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
```


running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdatePublicIpRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdatePublicIpRequestBody(
        elb_id="3ee8da68-5a23-4ca4-b5e0-74ff996e16be"
    )
    response = client.update_public_ip(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 开启Redis 3.0版本公网访问。

coding: utf-8

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdatePublicIpRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdatePublicIpRequestBody(
            enable_ssl=True,
            publicip_id="00130845-97c3-43fa-88eb-85cfb08d87ca"
        )
        response = client.update_public_ip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 开启Redis 4.0及以上版本公网访问。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePublicIpRequest{}
    request.InstanceId = "{instance_id}"
    elbIdUpdatePublicIpRequestBody := "3ee8da68-5a23-4ca4-b5e0-74ff996e16be"
    request.Body = &model.UpdatePublicIpRequestBody{
        ElbId: &elbIdUpdatePublicIpRequestBody,
    }
    response, err := client.UpdatePublicIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 开启Redis 3.0版本公网访问。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdatePublicIpRequest{}
request.InstanceId = "{instance_id}"
enableSslUpdatePublicIpRequestBody:= true
publicIpIdUpdatePublicIpRequestBody:= "00130845-97c3-43fa-88eb-85cfb08d87ca"
request.Body = &model.UpdatePublicIpRequestBody{
    EnableSsl: &enableSslUpdatePublicIpRequestBody,
    PublicIpId: &publicIpIdUpdatePublicIpRequestBody,
}
response, err := client.UpdatePublicIp(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	提交开启实例公网访问任务成功。
204	开启/修改 Redis 3.0 实例公网访问成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.15 关闭实例公网访问

功能介绍

关闭实例公网访问。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/public-ip

表 5-150 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200

表 5-151 响应 Body 参数

参数	参数类型	描述
job_id	String	请求提交任务ID。

状态码：204

关闭 Redis 3.0 实例公网访问成功。

状态码：400

表 5-152 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-153 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-154 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-155 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-156 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/public-ip
```

响应示例

状态码：200

提交关闭实例公网访问任务成功。

```
{
  "job_id" : "ff808081816ba43101816ba4e0ea0000"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4063",
  "error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeletePublicIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        DeletePublicIpRequest request = new DeletePublicIpRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeletePublicIpResponse response = client.deletePublicIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
```



```
.with_credentials(credentials) \
.with_region(DcsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeletePublicIpRequest()
    request.instance_id = "{instance_id}"
    response = client.delete_public_ip(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeletePublicIpRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.DeletePublicIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	提交关闭实例公网访问任务成功。
204	关闭 Redis 3.0 实例公网访问成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.16 集群分片倒换

功能介绍

集群分片倒换，适用于proxy和cluster实例。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instance/{instance_id}/groups/{group_id}/replications/{node_id}/async-switchover

表 5-157 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
group_id	是	String	参数解释： 分片ID。可通过DCS控制台进入节点管理界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
node_id	是	String	参数解释： 升级为主节点的节点ID。可通过DCS控制台进入节点管理界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码： 200

表 5-158 响应 Body 参数

参数	参数类型	描述
job_id	String	参数解释： 集群分片倒换的任务ID。 取值范围： 不涉及。

状态码：400

表 5-159 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-160 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-161 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-162 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-163 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/groups/{group_id}/replications/{node_id}/
async-switchover
```

响应示例

状态码：200

参数解释：

返回集群分片倒换的任务ID成功。

```
{
  "job_id" : "8abf6e1e87bce0740187bd32748b0530"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4063",
  "error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ExecuteClusterSwitchoverSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ExecuteClusterSwitchoverRequest request = new ExecuteClusterSwitchoverRequest();
        request.withInstanceId("{instance_id}");
        request.withGroupId("{group_id}");
        request.withNodeId("{node_id}");
        try {
            ExecuteClusterSwitchoverResponse response = client.executeClusterSwitchover(request);
            System.out.println(response.toString());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExecuteClusterSwitchoverRequest()
        request.instance_id = "{instance_id}"
        request.group_id = "{group_id}"
        request.node_id = "{node_id}"
        response = client.execute_cluster_switchover(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ExecuteClusterSwitchoverRequest{}
request.InstanceId = "{instance_id}"
request.GroupId = "{group_id}"
request.NodeId = "{node_id}"
response, err := client.ExecuteClusterSwitchover(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 返回集群分片倒换的任务ID成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.17 异步交换实例主备节点

功能介绍

异步交换实例主备节点

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/async-swap

表 5-164 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200

表 5-165 响应 Body 参数

参数	参数类型	描述
job_id	String	主备切换任务ID。

状态码：400

表 5-166 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-167 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-168 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-169 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-170 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/async-swap
```

响应示例

状态码：200

异步交换实例主备节点成功。

```
{
  "job_id" : "8abfa7aa88ad5aa50188aef7ec0936d1"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ChangeMasterStandbyAsyncSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        ChangeMasterStandbyAsyncRequest request = new ChangeMasterStandbyAsyncRequest();
        request.withInstanceId("{instance_id}");
        try {
            ChangeMasterStandbyAsyncResponse response = client.changeMasterStandbyAsync(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(DcsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ChangeMasterStandbyAsyncRequest()
    request.instance_id = "{instance_id}"
    response = client.change_master_standby_async(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeMasterStandbyAsyncRequest{
        request.InstanceId = "{instance_id}"
    }
    response, err := client.ChangeMasterStandbyAsync(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	异步交换实例主备节点成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.18 查询实例是否可以扩容

功能介绍

查询实例是否可以扩容。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/operations

表 5-171 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-172 Query 参数

参数	是否必选	参数类型	描述
operation	是	String	参数解释： 操作信息。 约束限制： 不涉及。 取值范围： extend：表示扩容。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-173 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。

参数	参数类型	描述
operations	Array of Operations objects	参数解释： 支持的操作列表，目前只支持扩容。

表 5-174 Operations

参数	参数类型	描述
operation	String	参数解释： 操作信息。 取值范围： 不涉及。
is_support	Boolean	参数解释： 是否支持该操作。 取值范围： true：支持。 false：不支持
cause_id	String	参数解释： 不支持该操作的错误原因。 取值范围： 不涉及。

状态码：400

表 5-175 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-176 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-177 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-178 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-179 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/operations?operation=extend
```

响应示例

状态码：200

查询实例是否可以扩容成功。

```
{
  "instance_id" : "dacf949d-7f8a-4244-a4e8-6a4119186ac3",
  "operations" : [ {
    "operation" : "extend",
    "is_support" : true,
    "cause_id" : null
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4063",
  "error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListInstanceOperationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstanceOperationsRequest request = new ListInstanceOperationsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListInstanceOperationsResponse response = client.listInstanceOperations(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListInstanceOperationsRequest()
    request.instance_id = "{instance_id}"
    response = client.list_instance_operations(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstanceOperationsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListInstanceOperations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例是否可以扩容成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.19 查询集群实例拓扑关系图

功能介绍

查询集群实例拓扑关系图。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/nodes

表 5-180 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-181 Query 参数

参数	是否必选	参数类型	描述
start	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示条数。 约束限制： 不涉及。 取值范围： 1~1000 默认取值： 10

请求参数

无

响应参数

状态码： 200

表 5-182 响应 Body 参数

参数	参数类型	描述
redis_server	Array of TopologyInfo objects	参数解释: 集群角色: redis-server实例拓扑图。
cluster_lvs	Array of TopologyInfo objects	参数解释: 集群角色: lvs实例拓扑图(当前已下线), 只适用于Redis 3.0版本实例。
cluster_admin	Array of TopologyInfo objects	参数解释: 集群角色: admin实例拓扑图(当前已下线), 只适用于Redis 3.0版本实例。
cluster_proxy	Array of TopologyInfo objects	参数解释: 集群角色: proxy实例拓扑图。
master	Array of TopologyInfo objects	参数解释: Redis的角色master实例拓扑图, 只适用于Redis 3.0版本实例。
vpcendpoint	Array of TopologyInfo objects	参数解释: 集群角色: VPC Endpoint服务实例拓扑图, 适用于4.0及以上版本实例。
elb	Array of TopologyInfo objects	参数解释: 集群角色: ELB服务实例拓扑图, 适用于Redis 4.0及以上版本实例。

表 5-183 TopologyInfo

参数	参数类型	描述
node_id	String	参数解释: 节点ID。 取值范围: 不涉及。
node_name	String	参数解释: 节点名称。 取值范围: 不涉及。
ip	String	参数解释: 节点IP。 取值范围: 不涉及。

参数	参数类型	描述
port	String	参数解释： 节点端口。 取值范围： 不涉及。
node_type	String	参数解释： 节点主从角色。 取值范围： master：主节点 slave：从节点 proxy：proxy节点
max_memory	String	参数解释： 总内存，单位：MB。 取值范围： 不涉及。
used_memory	String	参数解释： 已使用的内存，单位：MB。 取值范围： 不涉及。
qps	String	参数解释： 每秒查询率。 取值范围： 不涉及。
bandwidth	BandWidth object	参数解释： 带宽信息。
db_num	String	参数解释： 该实例的DB数量。 取值范围： 不涉及。
dbs	KeySpace object	参数解释： 键信息。
relation_ip	String	参数解释： 关联IP。 取值范围： 不涉及。

参数	参数类型	描述
relation_port	String	参数解释： 关联端口。 取值范围： 不涉及。
group_id	String	参数解释： 分片ID。 取值范围： 不涉及。
status	String	参数解释： 节点状态。 取值范围： 不涉及。
dims	DimsInfo object	参数解释： description: CES监控维护信息。

表 5-184 BandWidth

参数	参数类型	描述
input	String	参数解释： 上行带宽，单位kbit/s。 取值范围： 不涉及。
output	String	参数解释： 下行带宽，单位kbit/s。 取值范围： 不涉及。

表 5-185 KeySpace

参数	参数类型	描述
db_idx	String	参数解释： db索引值。 取值范围： 不涉及。

参数	参数类型	描述
keys	String	参数解释： 节点键数量。 取值范围： 不涉及。
expires	String	参数解释： 节点过期键数量。 取值范围： 不涉及。
avg_ttl	String	参数解释： 节点键的平均过期时间。 取值范围： 不涉及。

表 5-186 DimsInfo

参数	参数类型	描述
dim_k	String	参数解释： CES监控维度路由。 取值范围： 不涉及。
dim_v	String	参数解释： CES监控维度ID。 取值范围： 不涉及。

状态码：400

表 5-187 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-188 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-189 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-190 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-191 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/nodes
```

响应示例

状态码：200

查询实例拓扑成功。

```
{
  "cluster_proxy": [ {
    "node_id": "ff80808188b4aeb60188b4b234ea0144",
    "node_name": "dcs-bare-vm-x86-cn-north-7c-1657002433-0",
    "ip": "192.168.60.32",
    "port": "2892",
    "node_type": "proxy",
    "max_memory": null,
    "used_memory": null,
    "qps": null,
    "bandwidth": null,
    "db_num": null,
    "dbs": null,
    "relation_ip": null,
    "relation_port": null,
    "group_id": null,
    "status": "Active",
    "dims": {
      "dim_k": "dcs_instance_id",
      "dim_v": "3b76eb20-70cd-4083-be96-266a29de6e52"
    }
  }, {
    "node_id": "ff80808188b4aeb60188b4b234ea0143",
    "node_name": "dcs-bare-vm-x86-cn-north-7c-1684760402-1",
    "ip": "192.168.5.105",
    "port": "3048",
    "node_type": "proxy",
    "max_memory": null,
```



```
"used_memory" : null,
"qps" : null,
"bandwidth" : null,
"db_num" : null,
"dbs" : null,
"relation_ip" : null,
"relation_port" : null,
"group_id" : null,
"status" : "Active",
"dims" : {
  "dim_k" : "dcs_instance_id",
  "dim_v" : "3b76eb20-70cd-4083-be96-266a29de6e52"
}
}, {
  "node_id" : "ff80808188b4aeb60188b4b234ea0145",
  "node_name" : "dcs-bare-vm-x86-cn-north-7c-1660116188-2",
  "ip" : "192.168.53.35",
  "port" : "3834",
  "node_type" : "proxy",
  "max_memory" : null,
  "used_memory" : null,
  "qps" : null,
  "bandwidth" : null,
  "db_num" : null,
  "dbs" : null,
  "relation_ip" : null,
  "relation_port" : null,
  "group_id" : null,
  "status" : "Active",
  "dims" : {
    "dim_k" : "dcs_instance_id",
    "dim_v" : "3b76eb20-70cd-4083-be96-266a29de6e52"
  }
} ],
"vpcendpoint" : [ {
  "node_id" : "e50c0fda-1870-4bbc-ba4c-ace69504e305",
  "node_name" : "",
  "ip" : "192.168.74.33",
  "port" : "6379",
  "node_type" : "vpcendpoint",
  "max_memory" : null,
  "used_memory" : null,
  "qps" : null,
  "bandwidth" : null,
  "db_num" : null,
  "dbs" : null,
  "relation_ip" : null,
  "relation_port" : null,
  "group_id" : null,
  "status" : "Active",
  "dims" : null
} ],
"elb" : [ {
  "node_id" : "af04b606-241e-4099-8ae2-194e2b61ef63",
  "node_name" : "",
  "ip" : "192.168.20.137",
  "port" : "6379",
  "node_type" : "elb",
  "max_memory" : null,
  "used_memory" : null,
  "qps" : null,
  "bandwidth" : null,
  "db_num" : null,
  "dbs" : null,
  "relation_ip" : null,
  "relation_port" : null,
  "group_id" : null,
  "status" : "Active",
  "dims" : null
}
```

```

    }],
    "redis_server": [ {
      "node_id": "ff80808188b4aeb60188b4b234e4013d",
      "node_name": "dcs-bare-vm-x86-cn-north-7c-1684823334-0",
      "ip": "192.168.22.138",
      "port": "2076",
      "node_type": "master",
      "max_memory": "1365",
      "used_memory": "3",
      "qps": "4",
      "bandwidth": {
        "input": "0.10",
        "output": "0.30"
      },
      "db_num": "0",
      "dbs": {
        "db_idx": "0",
        "keys": "0",
        "expires": "0",
        "avg_ttl": "0"
      },
      "relation_ip": "192.168.60.32",
      "relation_port": "2984",
      "group_id": null,
      "status": "Active",
      "dims": {
        "dim_k": "dcs_instance_id",
        "dim_v": "3b76eb20-70cd-4083-be96-266a29de6e52"
      }
    }, {
      "node_id": "ff80808188b4aeb60188b4b234ea0141",
      "node_name": "dcs-bare-vm-x86-cn-north-7c-1657800431-0",
      "ip": "192.168.33.181",
      "port": "3694",
      "node_type": "master",
      "max_memory": "1365",
      "used_memory": "3",
      "qps": "8",
      "bandwidth": {
        "input": "0.20",
        "output": "3.31"
      },
      "db_num": "0",
      "dbs": {
        "db_idx": "0",
        "keys": "0",
        "expires": "0",
        "avg_ttl": "0"
      },
      "relation_ip": "192.168.46.161",
      "relation_port": "3600",
      "group_id": null,
      "status": "Active",
      "dims": {
        "dim_k": "dcs_instance_id",
        "dim_v": "3b76eb20-70cd-4083-be96-266a29de6e52"
      }
    }, {
      "node_id": "ff80808188b4aeb60188b4b234ea013f",
      "node_name": "dcs-bare-vm-x86-cn-north-7c-1657002433-0",
      "ip": "192.168.60.32",
      "port": "2476",
      "node_type": "master",
      "max_memory": "1365",
      "used_memory": "3",
      "qps": "3",
      "bandwidth": {
        "input": "0.12",
        "output": "2.30"
      }
    }
  ],
  "status": "Active",
  "group_id": null,
  "relation_ip": "192.168.60.32",
  "relation_port": "2984",
  "avg_ttl": "0",
  "keys": "0",
  "expires": "0",
  "qps": "4",
  "bandwidth": {
    "input": "0.10",
    "output": "0.30"
  },
  "db_num": "0",
  "dbs": {
    "db_idx": "0",
    "keys": "0",
    "expires": "0",
    "avg_ttl": "0"
  },
  "dim_k": "dcs_instance_id",
  "dim_v": "3b76eb20-70cd-4083-be96-266a29de6e52"
}

```

```
    },
    "db_num" : "0",
    "dbs" : {
      "db_idx" : "0",
      "keys" : "0",
      "expires" : "0",
      "avg_ttl" : "0"
    },
    "relation_ip" : "192.168.53.35",
    "relation_port" : "3704",
    "group_id" : null,
    "status" : "Active",
    "dims" : {
      "dim_k" : "dcs_instance_id",
      "dim_v" : "3b76eb20-70cd-4083-be96-266a29de6e52"
    }
  }
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowInstanceTopologySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceTopologyRequest request = new ShowInstanceTopologyRequest();
```

```
request.withInstanceId("{instance_id}");
try {
    ShowInstanceTopologyResponse response = client.showInstanceTopology(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceTopologyRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_topology(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceTopologyRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowInstanceTopology(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例拓扑成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.20 获取实例分片带宽

功能介绍

获取实例各个分片带宽。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/bandwidths

表 5-192 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-193 响应 Body 参数

参数	参数类型	描述
bandwidth	Integer	参数解释： 实例当前带宽(Mbit/s)。 取值范围： 不涉及。
max_bandwidth	Integer	参数解释： 实例最大带宽(Mbit/s)。 取值范围： 不涉及。
allow_modify	Boolean	参数解释： 是否支持调带宽。 取值范围： true：支持。 false：不支持。
group_bandwidths	Array of GroupBandwidthInfo objects	参数解释： 分片带宽列表。

表 5-194 GroupBandwidthInfo

参数	参数类型	描述
group_id	String	参数解释： 分片ID。 取值范围： 不涉及。
updated_at	String	参数解释： 更新时间，UTC时间。 取值范围： 不涉及。
bandwidth	Integer	参数解释： 当前带宽(Mbit/s)。 取值范围： 不涉及。

参数	参数类型	描述
max_bandwidth	Integer	参数解释： 最大带宽(Mbit/s)。 取值范围： 不涉及。
assured_bandwidth	Integer	参数解释： 基准带宽(Mbit/s)。 取值范围： 不涉及。

状态码：400

表 5-195 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-196 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-197 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-198 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-199 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/bandwidths
```

响应示例

状态码：200

参数解释：

实例分片带宽。

```
{
  "bandwidth" : 900,
  "max_bandwidth" : 6000,
  "allow_modify" : true,
  "group_bandwidths" : [ {
    "group_id" : "6e19e25d-44fb-4a0b-bd46-3046baa30027",
    "updated_at" : null,
    "bandwidth" : 300,
    "max_bandwidth" : 2000,
    "assured_bandwidth" : 200
  }, {
    "group_id" : "1a64e15b-b4a4-4de8-8f35-5fb4faae2460",
    "updated_at" : null,
    "bandwidth" : 300,
    "max_bandwidth" : 2000,
    "assured_bandwidth" : 200
  }, {
    "group_id" : "e407d2bf-893a-4e8e-8b83-3ab3c0f7d755",
    "updated_at" : null,
    "bandwidth" : 300,
    "max_bandwidth" : 2000,
    "assured_bandwidth" : 200
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowBandwidthsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowBandwidthsRequest request = new ShowBandwidthsRequest();
request.withInstanceId("{instance_id}");
try {
    ShowBandwidthsResponse response = client.showBandwidths(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBandwidthsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_bandwidths(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
dc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBandwidthsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowBandwidths(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 实例分片带宽。
400	非法请求。
401	认证非法。
403	请求被拒绝。
404	路径不存在。
500	内部错误。

错误码

请参见[错误码](#)。

5.2.21 修改实例分片带宽

功能介绍

修改实例分片带宽。只有运行中状态的实例支持调整带宽。企业版Redis暂不支持调整带宽。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/bandwidths

表 5-200 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-201 请求 Body 参数

参数	是否必选	参数类型	描述
group_bandwidths	否	Array of UpdateGroupBandwidthInfoRequest objects	分片带宽列表。

表 5-202 UpdateGroupBandwidthInfoRequest

参数	是否必选	参数类型	描述
group_id	是	String	分片ID。
target_bandwidth	是	Integer	分片的目标带宽（Mbit/s）。实例单分片带宽的调整范围在单分片的默认基准带宽到2048 Mbit/s之间。 如果输入的目标带宽小于基准带宽，那么会调整为基准值。

响应参数

状态码：200

表 5-203 响应 Body 参数

参数	参数类型	描述
job_id	String	修改带宽后台任务ID

状态码：400

表 5-204 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-205 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-206 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-207 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-208 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/bandwidths
{
```

```
"group_bandwidths" : [ {  
  "group_id" : "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",  
  "target_bandwidth" : 500  
} ]  
}
```

响应示例

状态码：200

提交带宽任务成功。

```
{  
  "job_id" : "ff80808188bfde180188c223434d6f7a"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateBandwidthSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateBandwidthRequest request = new UpdateBandwidthRequest();  
        request.withInstanceId("{instance_id}");  
        UpdateBandwidthInfoRequest body = new UpdateBandwidthInfoRequest();  
        List<UpdateGroupBandwidthInfoRequest> listbodyGroupBandwidths = new ArrayList<>();  
        listbodyGroupBandwidths.add(  
            new UpdateGroupBandwidthInfoRequest()  
                .withGroupId("2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef")  
                .withTargetBandwidth(500)  
        );  
        body.withGroupBandwidths(listbodyGroupBandwidths);  
    }  
}
```

```
request.withBody(body);
try {
    UpdateBandwidthResponse response = client.updateBandwidth(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateBandwidthRequest()
        request.instance_id = "{instance_id}"
        listGroupBandwidthsbody = [
            UpdateGroupBandwidthInfoRequest(
                group_id="2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",
                target_bandwidth=500
            )
        ]
        request.body = UpdateBandwidthInfoRequest(
            group_bandwidths=listGroupBandwidthsbody
        )
        response = client.update_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateBandwidthRequest{}
    request.InstanceId = "{instance_id}"
    var listGroupBandwidthsbody = []model.UpdateGroupBandwidthInfoRequest{
        {
            GroupId: "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef",
            TargetBandwidth: int32(500),
        },
    }
    request.Body = &model.UpdateBandwidthInfoRequest{
        GroupBandwidths: &listGroupBandwidthsbody,
    }
    response, err := client.UpdateBandwidth(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	提交带宽任务成功。
400	非法请求。
401	认证非法。
403	请求被拒绝。

状态码	描述
404	路径不存在。
500	内部错误。

错误码

请参见[错误码](#)。

5.2.22 异步导出实例资源

功能介绍

调用本接口，返回的任务ID（job_id）用于[查询实例列表导出任务详情](#)接口，获取导出的实例列表。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/export

表 5-209 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-210 请求 Body 参数

参数	是否必选	参数类型	描述
included_instances	否	Array of strings	导出实例列表，如果为空，则会导出满足其余参数条件的所有实例。
region	是	String	局点名称，用于导出的文件名命名。
name	否	String	按照实例名称筛选实例。
capacity	否	String	按照实例容量筛选实例。
instance_id	否	String	按照实例ID筛选实例。

参数	是否必选	参数类型	描述
ip	否	String	按照IP地址筛选实例。
available_zone	否	String	按照可用区筛选实例。
status	否	String	按照实例状态筛选实例，取值范围如下： • RUNNING：运行中 • EXTENDING：扩缩容中 • CREATING：创建中 • ERROR：异常状态 • DELETEFAILED：删除失败 • FROZEN：冻结状态 • RESTARTING：重启中
product_type	否	String	按照产品类型筛选实例，取值范围如下： • generic：基础版 • enterprise：企业版
cache_mode	否	String	按照实例类型筛选实例，取值范围如下： • single：单机实例 • ha：主备实例 • ha_rw_split：读写分离实例 • cluster：cluster集群实例 • proxy：proxy集群实例
engine	否	String	按照缓存引擎筛选实例，当前仅支持Redis。
engine_version	否	String	按照缓存引擎版本筛选实例，例如：4.0。
cpu_type	否	String	按照CPU类型筛选实例，分为x86_64和aarch64。
enterprise_project_id	否	String	按照企业项目ID筛选实例。
charging_mode	否	String	按照计费方式筛选实例。
tags	否	String	按照标签筛选实例。

响应参数

状态码：200

表 5-211 响应 Body 参数

参数	参数类型	描述
job_id	String	导出实例的任务ID。

状态码：400

表 5-212 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-213 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-214 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-215 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-216 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/export
{
```

```
"included_instances" : [ ],
"region" : "Region",
"name" : "name",
"capacity" : "2",
"instance_id" : "52f265a5-13c4-42cc-b0fa-38ef830e6391",
"ip" : "192.168.78.24",
"available_zone" : "d539378ec1314c85b76fefa3f7071458",
"status" : "RUNNING",
"product_type" : "generic",
"cache_mode" : "proxy",
"engine" : "Redis",
"engine_version" : "4.0",
"cpu_type" : "x86",
"enterprise_project_id" : "93085bd3-6a84-4acc-bc70-5ebfa18c77b6",
"charging_mode" : "0",
"tags" : "aaa=cccc"
}
```

响应示例

状态码：200

异步导出实例资源返回成功。

```
{
  "job_id" : "ff80808188bfde180188c223434d6f7a"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ExportInstancesTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
```

```
.withRegion(DcsRegion.valueOf("<YOUR REGION>"))
.build();
ExportInstancesTaskRequest request = new ExportInstancesTaskRequest();
ExportInstancesTaskBody body = new ExportInstancesTaskBody();
body.withTags("aaa=cccc");
body.withChargingMode("0");
body.withEnterpriseProjectId("93085bd3-6a84-4acc-bc70-5ebfa18c77b6");
body.withCpuType("x86");
body.withEngineVersion("4.0");
body.withEngine("Redis");
body.withCacheMode("proxy");
body.withProductType(ExportInstancesTaskBody.ProductTypeEnum.fromValue("generic"));
body.withStatus("RUNNING");
body.withAvailableZone("d539378ec1314c85b76fefa3f7071458");
body.withIp("192.168.78.24");
body.withInstanceId("52f265a5-13c4-42cc-b0fa-38ef830e6391");
body.withCapacity("2");
body.withName("name");
body.withRegion("Region");
request.withBody(body);
try {
    ExportInstancesTaskResponse response = client.exportInstancesTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ExportInstancesTaskRequest()
        request.body = ExportInstancesTaskBody(
            tags="aaa=cccc",
            charging_mode="0",
```

```
enterprise_project_id="93085bd3-6a84-4acc-bc70-5ebfa18c77b6",
cpu_type="x86",
engine_version="4.0",
engine="Redis",
cache_mode="proxy",
product_type="generic",
status="RUNNING",
available_zone="d539378ec1314c85b76fefa3f7071458",
ip="192.168.78.24",
instance_id="52f265a5-13c4-42cc-b0fa-38ef830e6391",
capacity="2",
name="name",
region="Region"
)
response = client.export_instances_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportInstancesTaskRequest{
        tagsExportInstancesTaskBody:= "aaa=cccc"
        chargingModeExportInstancesTaskBody:= "0"
        enterpriseProjectIdExportInstancesTaskBody:= "93085bd3-6a84-4acc-bc70-5ebfa18c77b6"
        cpuTypeExportInstancesTaskBody:= "x86"
        engineVersionExportInstancesTaskBody:= "4.0"
        engineExportInstancesTaskBody:= "Redis"
        cacheModeExportInstancesTaskBody:= "proxy"
        productTypeExportInstancesTaskBody:= model.GetExportInstancesTaskBodyProductTypeEnum().GENERIC
        statusExportInstancesTaskBody:= "RUNNING"
        availableZoneExportInstancesTaskBody:= "d539378ec1314c85b76fefa3f7071458"
        ipExportInstancesTaskBody:= "192.168.78.24"
        instanceIdExportInstancesTaskBody:= "52f265a5-13c4-42cc-b0fa-38ef830e6391"
```

```
capacityExportInstancesTaskBody:= "2"
nameExportInstancesTaskBody:= "name"
request.Body = &model.ExportInstancesTaskBody{
    Tags: &tagsExportInstancesTaskBody,
    ChargingMode: &chargingModeExportInstancesTaskBody,
    EnterpriseProjectId: &enterpriseProjectIdExportInstancesTaskBody,
    CpuType: &cpuTypeExportInstancesTaskBody,
    EngineVersion: &engineVersionExportInstancesTaskBody,
    Engine: &engineExportInstancesTaskBody,
    CacheMode: &cacheModeExportInstancesTaskBody,
    ProductType: &productTypeExportInstancesTaskBody,
    Status: &statusExportInstancesTaskBody,
    AvailableZone: &availableZoneExportInstancesTaskBody,
    Ip: &ipExportInstancesTaskBody,
    InstanceId: &instanceIdExportInstancesTaskBody,
    Capacity: &capacityExportInstancesTaskBody,
    Name: &nameExportInstancesTaskBody,
    Region: "Region",
}
response, err := client.ExportInstancesTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	异步导出实例资源返回成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.23 指定实例节点启停开关

功能介绍

实例节点启停。执行节点关机操作前的24小时内，需要对实例（单机实例除外）进行数据备份。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/nodes/status

表 5-217 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-218 请求 Body 参数

参数	是否必选	参数类型	描述
node_ids	否	Array of strings	实例需要启停的节点ID列表。未配置该参数时，默认启停实例全部节点。
action	否	String	对实例节点的操作： start：开启节点 stop：关闭节点

响应参数

状态码：200

表 5-219 响应 Body 参数

参数	参数类型	描述
job_id	String	实例节点启停的任务ID。

状态码：400

表 5-220 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-221 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-222 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-223 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：405

表 5-224 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-225 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/nodes/status
{
```

```
"action" : "start",  
"node_ids" : [ "3e803f66-fbb0-47ad-b6cb-fb87f5bed4aa", "3eaaff66-fbb0-47ad-b6cb-fb87f5b12fbb" ]  
}
```

响应示例

状态码：200

实例节点启停任务下发成功。

```
{  
  "job_id" : "8abfa7aa88ad5aa50188aef7ec0936d1"  
}
```

状态码：405

当前请求不被允许。

```
{  
  "error_code" : "DCS.2001",  
  "error_msg" : "the latest backup records over 24h old, need to create new backup file."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ChangeNodesStartStopStatusSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ChangeNodesStartStopStatusRequest request = new ChangeNodesStartStopStatusRequest();  
        request.withInstanceId("{instance_id}");  
        ChangeNodesStartStopStatusBody body = new ChangeNodesStartStopStatusBody();
```

```
List<String> listbodyNodeIds = new ArrayList<>();
listbodyNodeIds.add("3e803f66-fbb0-47ad-b6cb-fb87f5bed4aa");
listbodyNodeIds.add("3eaaff66-fbb0-47ad-b6cb-fb87f5b12fbb");
body.withAction("start");
body.withNodeIds(listbodyNodeIds);
request.withBody(body);
try {
    ChangeNodesStartStopStatusResponse response = client.changeNodesStartStopStatus(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeNodesStartStopStatusRequest()
        request.instance_id = "{instance_id}"
        listNodeIdsbody = [
            "3e803f66-fbb0-47ad-b6cb-fb87f5bed4aa",
            "3eaaff66-fbb0-47ad-b6cb-fb87f5b12fbb"
        ]
        request.body = ChangeNodesStartStopStatusBody(
            action="start",
            node_ids=listNodeIdsbody
        )
        response = client.change_nodes_start_stop_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeNodesStartStopStatusRequest{}
    request.InstanceId = "{instance_id}"
    var listNodeIdsbody = []string{
        "3e803f66-fbb0-47ad-b6cb-fb87f5bed4aa",
        "3eaaff66-fbb0-47ad-b6cb-fb87f5b12fbb",
    }
    actionChangeNodesStartStopStatusBody := "start"
    request.Body = &model.ChangeNodesStartStopStatusBody{
        Action: &actionChangeNodesStartStopStatusBody,
        NodeIds: &listNodeIdsbody,
    }
    response, err := client.ChangeNodesStartStopStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	实例节点启停任务下发成功。
400	非法请求。

状态码	描述
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
405	当前请求不被允许。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.24 根据实例 ID 获取实例内核版本信息

功能介绍

获取对应实例内核版本号。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/version

表 5-226 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-227 响应 Body 参数

参数	参数类型	描述
engine_minor_version	String	参数解释： 当前实例内核小版本信息。 取值范围： 不涉及。
latest_engine_minor_version	String	参数解释： DCS实例最新的内核小版本信息。 取值范围： 不涉及。
proxy_minor_version	String	参数解释： 当前实例proxy代理节点版本信息。 取值范围： 不涉及。
latest_proxy_minor_version	String	参数解释： DCS实例最新的proxy代理节点版本信息。 取值范围： 不涉及。

参数	参数类型	描述
engine_minor_version_upgradable	Boolean	参数解释： 当前实例内核是否可以升级。 取值范围： true：可以升级。 false：已经是最新版本或当前版本不支持升级，无需升级。
proxy_minor_version_upgradable	Boolean	参数解释： 当前实例proxy代理节点是否可以升级。 取值范围： true：可以升级。 false：已经是最新版本或当前版本不支持升级，无需升级。

状态码：400

表 5-228 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-229 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-230 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-231 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-232 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/version
```

响应示例

状态码：200

参数解释：

返回对应实例版本信息内容。

```
{
  "engine_minor_version": "6.0.1.0",
  "proxy_minor_version": "5.0.14.0",
  "latest_proxy_minor_version": "5.0.14.0",
  "latest_engine_minor_version": "6.0.1.0",
  "engine_minor_version_upgradable": false,
  "proxy_minor_version_upgradable": false
}
```

状态码：400

参数解释：

非法请求。

```
{
  "error_code": "DCS.4063",
  "error_msg": "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowInstanceVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
```

```
ShowInstanceVersionRequest request = new ShowInstanceVersionRequest();
request.withInstanceId("{instance_id}");
try {
    ShowInstanceVersionResponse response = client.showInstanceVersion(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceVersionRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceVersionRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 返回对应实例版本信息内容。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.2.25 升级实例小版本/代理版本

功能介绍

升级实例小版本/代理版本。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/minor-version/upgrade

表 5-233 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-234 请求 Body 参数

参数	是否必选	参数类型	描述
proxy_minor_version	否	String	Proxy代理节点目标版本号，设置为latest时，即升级到最新版本。
engine_minor_version	否	String	引擎目标小版本号，设置为latest时，即升级到最新版本。

响应参数

状态码：200

表 5-235 响应 Body 参数

参数	参数类型	描述
task_id	String	后台任务ID。

状态码：400

表 5-236 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-237 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-238 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-239 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-240 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
{
  "proxy_minor_version" : "5.0.14.1",
  "engine_minor_version" : "latest"
}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4062",
  "error_msg" : "Invalid instance_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```



```
public class UpgradeInstanceMinorVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpgradeInstanceMinorVersionRequest request = new UpgradeInstanceMinorVersionRequest();
        request.withInstanceId("{instance_id}");
        UpgradeMinorVersionRequestBody body = new UpgradeMinorVersionRequestBody();
        body.withEngineMinorVersion("latest");
        body.withProxyMinorVersion("5.0.14.1");
        request.withBody(body);
        try {
            UpgradeInstanceMinorVersionResponse response = client.upgradeInstanceMinorVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = UpgradeInstanceMinorVersionRequest()
    request.instance_id = "{instance_id}"
    request.body = UpgradeMinorVersionRequestBody(
        engine_minor_version="latest",
        proxy_minor_version="5.0.14.1"
    )
    response = client.upgrade_instance_minor_version(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpgradeInstanceMinorVersionRequest{}
    request.InstanceId = "{instance_id}"
    engineMinorVersionUpgradeMinorVersionRequestBody := "latest"
    proxyMinorVersionUpgradeMinorVersionRequestBody := "5.0.14.1"
    request.Body = &model.UpgradeMinorVersionRequestBody{
        EngineMinorVersion: &engineMinorVersionUpgradeMinorVersionRequestBody,
        ProxyMinorVersion: &proxyMinorVersionUpgradeMinorVersionRequestBody,
    }
    response, err := client.UpgradeInstanceMinorVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	提交升级任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.26 查询实例重命名的命令

功能介绍

查询实例重命名的命令。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/rename-commands

表 5-241 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-242 响应 Body 参数

参数	参数类型	描述
rename_commands	Array of strings	参数解释： 实例节点重命名的命令列表。

状态码：400

表 5-243 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-244 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-245 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-246 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-247 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/rename-commands
```

响应示例

状态码：200

查询实例重命名的命令成功。

```
{
  "rename_commands": [ "keys", "flushall" ]
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4062",
  "error_msg": "Invalid instance_id parameter in the request."
}
```

状态码

状态码	描述
200	查询实例重命名的命令成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。

状态码	描述
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.2.27 切换指定实例的子网信息

功能介绍

实例子网切换。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/resources/subnet

表 5-248 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-249 请求 Body 参数

参数	是否必选	参数类型	描述
action	否	String	参数解释： 实例切换VPC子网 约束限制： 不涉及。 取值范围： switch：标识子网切换 默认取值： 不涉及。
vpc_id	否	String	参数解释： 目标VPC ID 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
subnet_id	否	String	参数解释： 目标子网ID 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-250 响应 Body 参数

参数	参数类型	描述
job_id	String	参数解释： 实例子网切换下发任务ID 取值范围： 不涉及。

状态码：400

表 5-251 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-252 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-253 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-254 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-255 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/resources/subnet
{
  "action": "switch",
  "vpc_id": "c0a0a222-bac8-45d9-9d3e-9e7d0f9177c9",
  "subnet_id": "3e803f66-fbb0-47ad-b6cb-fb87f5bed4aa"
}
```

响应示例

状态码：200

参数解释：

实例子网切换任务下发成功。

```
{
  "job_id": "8abfa7aa88ad5aa50188aef7ec0936d1"
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4062",
  "error_msg": "Invalid subnet_id parameter in the request."
}
```

状态码

状态码	描述
200	参数解释： 实例子网切换任务下发成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.3 分片与副本

5.3.1 设置备节点优先级

功能介绍

设置副本优先级，主节点故障时，权重越小的备节点切换为主节点的优先级越高。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/groups/{group_id}/replications/{node_id}/slave-priority

表 5-256 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
group_id	是	String	分片ID。
node_id	是	String	节点ID。

请求参数

表 5-257 请求 Body 参数

参数	是否必选	参数类型	描述
slave_priority_weight	是	Integer	副本优先级，取值范围是0到100，0为默认禁止倒换。

响应参数

状态码：204
设置副本优先级成功。

状态码：400

表 5-258 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-259 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-260 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-261 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-262 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

设置实例备节点优先级为100。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/groups/{group_id}/replications/{node_id}/slave-priority
{
  "slave_priority_weight" : 100
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

设置实例备节点优先级为100。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateSlavePrioritySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSlavePriorityRequest request = new UpdateSlavePriorityRequest();
        request.withInstanceId("{instance_id}");
        request.withGroupId("{group_id}");
        request.withNodeId("{node_id}");
        PriorityBody body = new PriorityBody();
        body.withSlavePriorityWeight(100);
        request.withBody(body);
        try {
            UpdateSlavePriorityResponse response = client.updateSlavePriority(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

设置实例备节点优先级为100。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSlavePriorityRequest()
        request.instance_id = "{instance_id}"
        request.group_id = "{group_id}"
        request.node_id = "{node_id}"
        request.body = PriorityBody(
            slave_priority_weight=100
        )
        response = client.update_slave_priority(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

设置实例备节点优先级为100。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSlavePriorityRequest{
        request.InstanceId = "{instance_id}"
        request.GroupId = "{group_id}"
        request.NodeId = "{node_id}"
        request.Body = &model.PriorityBody{
            SlavePriorityWeight: int32(100),
        }
    }
    response, err := client.UpdateSlavePriority(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	设置副本优先级成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.3.2 域名摘除 IP

功能介绍

将只读副本的IP从域名中摘除，摘除成功后，只读域名不会再解析到该副本IP。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/groups/{group_id}/replications/{node_id}/remove-ip

表 5-263 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
group_id	是	String	分片ID。
node_id	是	String	节点ID。

请求参数

无

响应参数

状态码：200

表 5-264 响应 Body 参数

参数	参数类型	描述
job_id	String	域名摘除IP的任务ID。

状态码：400

表 5-265 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-266 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-267 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-268 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-269 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/groups/{group_id}/replications/{node_id}/remove-ip
```

响应示例

无

状态码

状态码	描述
200	域名摘除IP成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.3.3 查询分片信息

功能介绍

查询Redis 4.0及以上版本实例的分片和副本信息，单机实例不支持。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instance/{instance_id}/groups

表 5-270 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-271 响应 Body 参数

参数	参数类型	描述
group_list	Array of InstanceGroupListInfo objects	参数解释： 实例分片列表。 取值范围： 不涉及。
group_count	Integer	参数解释： 实例分片总数。 取值范围： 不涉及。

表 5-272 InstanceGroupListInfo

参数	参数类型	描述
group_id	String	参数解释： 实例分片ID。 取值范围： 不涉及。
group_name	String	参数解释： 实例分片名称。 取值范围： 不涉及。
replication_list	Array of InstanceReplicationListInfo objects	参数解释： 每个分片包含的副本列表。 取值范围： 不涉及。

表 5-273 InstanceReplicationListInfo

参数	参数类型	描述
replication_role	String	参数解释： 副本角色。 取值范围： - master：表示主节点。 - slave：表示从节点。

参数	参数类型	描述
replication_ip	String	参数解释： 副本IP。 取值范围： 不涉及。
is_replication	Boolean	参数解释： 是否为新增副本。 取值范围： • true • false
replication_id	String	参数解释： 副本ID。 取值范围： 不涉及。
node_id	String	参数解释： 节点ID。 取值范围： 不涉及。
status	String	参数解释： 副本状态。 取值范围： 不涉及。
az_code	String	参数解释： 副本所在的可用区。 取值范围： 不涉及。

参数	参数类型	描述
dimensions	Array of InstanceReplicationDimensionsInfo objects	参数解释： 副本对应的监控指标维度信息。可用于调用云监控服务的查询监控数据指标相关接口。 • 副本的监控维度为多维度，返回数组中包含两个维度信息。从云监控查询监控数据时，要按多维度传递指标维度参数，才能查询到监控指标值。 • 第一个维度为副本父维度信息，维度名称为dcs_instance_id，维度值对应副本所在的实例ID。 • 第二个维度，维度名称为dcs_cluster_redis_node，维度值为副本的监控对象ID，与副本ID和节点ID不同。 取值范围： 不涉及。

表 5-274 InstanceReplicationDimensionsInfo

参数	参数类型	描述
name	String	参数解释： 监控维度名称。 取值范围： • dcs_instance_id：实例维度。 • dcs_cluster_redis_node：数据节点维度。
value	String	参数解释： 维度取值。 取值范围： 不涉及。

状态码：500

表 5-275 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：400

表 5-276 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/groups
```

响应示例

状态码：200

查询分片和副本信息成功。

```
{
  "group_list": [ {
    "group_id": "35e1bed6-7de5-4898-9eb2-c362c783df15",
    "group_name": "group-0",
    "replication_list": [ {
      "az_code": "region01",
      "is_replication": false,
      "node_id": "8abfa7aa73291f15017329e1cda0002d",
      "replication_id": "dfbd3f84-08bc-42f0-b538-01d03e6dc178",
      "replication_ip": "192.168.76.25",
      "replication_role": "master",
      "status": "Active",
      "dimensions": [ {
        "name": "dcs_instance_id",
        "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
      }, {
        "name": "dcs_cluster_redis_node",
        "value": "8263dc69629c5b2d840e9816fa9c6200"
      } ]
    }, {
      "az_code": "region01",
      "is_replication": false,
      "node_id": "8abfa7aa73291f15017329e1cdb0002e",
      "replication_id": "63d4c880-7050-464f-ab19-c8a297474d7d",
      "replication_ip": "192.168.78.207",
      "replication_role": "slave",
      "status": "Active",
      "dimensions": [ {
        "name": "dcs_instance_id",
        "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
      }, {
        "name": "dcs_cluster_redis_node",
        "value": "8263dc69629c5b2d840e9816fa9c6201"
      } ]
    } ]
  }, {
    "group_id": "579a281f-6e63-4822-b0c7-e45c44b7c807",
    "group_name": "group-1",
    "replication_list": [ {
      "az_code": "region01",
      "is_replication": false,
      "node_id": "8abfa7aa73291f15017329e1cddb002f",
      "replication_id": "6284c192-48d1-462b-8fd9-45dad067c1a2",
      "replication_ip": "192.168.73.164",
      "replication_role": "master",
      "status": "Active",
      "dimensions": [ {
        "name": "dcs_instance_id",
        "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
      }, {
        "name": "dcs_cluster_redis_node",
        "value": "8263dc69629c5b2d840e9816fa9c6202"
      } ]
    }, {
      "az_code": "region01",
      "is_replication": false,
      "node_id": "8abfa7aa73291f15017329e1cdc80030",
      "replication_id": "b927de3b-42f3-45b5-b0e4-8547f0ef6727",
      "replication_ip": "192.168.77.172",
      "replication_role": "slave",
      "status": "Active",
      "dimensions": [ {
        "name": "dcs_instance_id",
```

```
    "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
  }, {
    "name": "dcs_cluster_redis_node",
    "value": "8263dc69629c5b2d840e9816fa9c6203"
  }
]
}, {
  "group_id": "c17305c6-6651-42d9-86bf-5a6087076eb7",
  "group_name": "group-2",
  "replication_list": [ {
    "az_code": "region01",
    "is_replication": false,
    "node_id": "8abfa7aa73291f15017329e1cdd90031",
    "replication_id": "caa6636d-a5c1-43b8-990a-3dc134da4522",
    "replication_ip": "192.168.76.143",
    "replication_role": "master",
    "status": "Active",
    "dimensions": [ {
      "name": "dcs_instance_id",
      "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
    }
  ], {
    "name": "dcs_cluster_redis_node",
    "value": "8263dc69629c5b2d840e9816fa9c6204"
  }
]
}, {
  "az_code": "region01",
  "is_replication": false,
  "node_id": "8abfa7aa73291f15017329e1cde50032",
  "replication_id": "4f46790d-a0b0-4a1b-aa02-1c554fccf62d",
  "replication_ip": "192.168.72.66",
  "replication_role": "slave",
  "status": "Active",
  "dimensions": [ {
    "name": "dcs_instance_id",
    "value": "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
  }
], {
  "name": "dcs_cluster_redis_node",
  "value": "8263dc69629c5b2d840e9816fa9c6205"
}
]
}],
"group_count": 3
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListGroupReplicationInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ListGroupReplicationInfoRequest request = new ListGroupReplicationInfoRequest();
request.withInstanceId("{instance_id}");
try {
    ListGroupReplicationInfoResponse response = client.listGroupReplicationInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListGroupReplicationInfoRequest()
        request.instance_id = "{instance_id}"
        response = client.list_group_replication_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```



```
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListGroupReplicationInfoRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListGroupReplicationInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询分片和副本信息成功。
500	内部服务错误。
400	非法请求。

错误码

请参见[错误码](#)。

5.3.4 批量查询实例节点信息

功能介绍

批量查询指定项目所有实例的节点信息、有效实例个数及节点个数。
创建中实例不返回节点信息。
支持Redis4.0及以上版本实例查询。

接口约束

支持Redis4.0及以上版本实例查询。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances-logical-nodes

表 5-277 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-278 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-100 默认取值： 10

请求参数

无

响应参数

状态码：200

表 5-279 响应 Body 参数

参数	参数类型	描述
count	Integer	参数解释： 查询结果的实例总数。 取值范围： 不涉及。
instances	Array of InstanceNodesInfoResp objects	参数解释： 实例列表。 取值范围： 不涉及。

表 5-280 InstanceNodesInfoResp

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
node_count	Integer	参数解释： 当前实例节点总数。 取值范围： 不涉及。
nodes	Array of NodesInfoResp objects	参数解释： 节点详情。 取值范围： 不涉及。

表 5-281 NodesInfoResp

参数	参数类型	描述
logical_node_id	String	参数解释： 逻辑节点ID。 取值范围： 不涉及。
name	String	参数解释： 节点名称。 取值范围： 不涉及。
status	String	参数解释： 节点状态。 取值范围： - Creating：创建中。 - Active：运行中。 - Inactive：故障。 - Deleting：删除中。 - AddSharding：添加分片中。

参数	参数类型	描述
az_code	String	参数解释： 可用区code。 取值范围： 不涉及。
node_role	String	参数解释： 节点角色。 取值范围： - redis-server: Redis server节点。 - redis-proxy: proxy节点。
node_type	String	参数解释： 节点主从角色。 取值范围： - master: 主节点。 - slave: 从节点。 - proxy: proxy实例节点角色为"proxy"。
node_ip	String	参数解释： 节点的IP。 取值范围： 不涉及。
node_port	String	参数解释： 节点的端口。 取值范围： 不涉及。
node_id	String	参数解释： 节点ID。 取值范围： 不涉及。
priority_weight	Integer	参数解释： 节点主备切换优先级权重。 取值范围： 0-100（1-100优先级逐步降低，1为最高，100为最低，0为禁止倒换）。

参数	参数类型	描述
is_access	Boolean	参数解释： 节点的IP是否可直接访问。 取值范围： • true • false
group_id	String	参数解释： 实例分片ID。 取值范围： 不涉及。
group_name	String	参数解释： 实例分片名称。 取值范围： 不涉及。
is_remove_ip	Boolean	参数解释： 是否从只读域名中摘除IP。 取值范围： • true • false
replication_id	String	参数解释： 实例副本ID。 取值范围： 不涉及。
dimensions	Array of InstanceReplicationDimensionsInfo objects	参数解释： 副本对应的监控指标维度信息。可用于调用云监控服务的查询监控数据指标相关接口。 • 副本的监控维度为多维度，返回数组中包含两个维度信息。从云监控查询监控数据时，要按多维度传递指标维度参数，才能查询到监控指标值。 • 第一个维度为副本父维度信息。维度名称为dcs_instance_id，维度值对应副本所在的实例ID。 • 第二个维度，维度名称为dcs_cluster_redis_node，维度值为副本的监控对象ID，与副本ID和节点ID不同。 取值范围： 不涉及。

表 5-282 InstanceReplicationDimensionsInfo

参数	参数类型	描述
name	String	参数解释： 监控维度名称。 取值范围： • dcs_instance_id：实例维度。 • dcs_cluster_redis_node：数据节点维度。
value	String	参数解释： 维度取值。 取值范围： 不涉及。

状态码：500

表 5-283 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：400

表 5-284 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances-logical-nodes?offset={offset}&limit={limit}
```

响应示例

状态码：200

查询节点信息成功。

```
{
  "count" : 1,
  "instances" : [ {
    "instance_id" : "b4468cb1-4f75-4866-a49d-86959acc0c9c",
    "node_count" : 2,
    "nodes" : [ {
      "logical_node_id" : "8f997eb512a92834bf5914be52ca8919",
      "name" : "(replica)192.168.0.178",
      "status" : "Active",
      "az_code" : "region01",
      "node_role" : "redis-server",
      "node_type" : "slave",
      "node_ip" : "192.168.0.178",
      "node_port" : "6379",
      "node_id" : "8a2c26c17f4d6516017f4d7b77940030",
      "is_access" : true,
      "group_id" : "ace962d1-7841-4e50-9313-c239fd75dee3",
      "group_name" : "group-0",
      "priority_weight" : 100,
      "is_remove_ip" : false,
      "replication_id" : "e40aa09e-03c2-4319-86eb-5ef4ca637544",
      "dimensions" : [ {
        "name" : "dcs_instance_id",
        "value" : "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
      }, {
        "name" : "dcs_cluster_redis_node",
        "value" : "8263dc69629c5b2d840e9816fa9c6200"
      } ]
    } ]
  } ]
}
```



```
    }}  
  }}  
}
```

状态码：500

内部服务错误。

```
{  
  "error_code" : "DCS.5000",  
  "error_msg" : "Internal service error."  
}
```

状态码：400

非法请求。

```
{  
  "error_code" : "DCS.4800",  
  "error_msg" : "Invalid offset in the request."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
public class BatchShowNodesInformationSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        BatchShowNodesInformationRequest request = new BatchShowNodesInformationRequest();  
        try {  
            BatchShowNodesInformationResponse response = client.batchShowNodesInformation(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  

```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchShowNodesInformationRequest()
        response = client.batch_show_nodes_information(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchShowNodesInformationRequest{}
response, err := client.BatchShowNodesInformation(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询节点信息成功。
500	内部服务错误。
400	非法请求。

错误码

请参见[错误码](#)。

5.3.5 查询实例节点信息

功能介绍

- 查询指定实例的节点信息。
- 支持Redis4.0及以上版本实例查询。
- 创建中的实例不返回节点信息。

接口约束

支持Redis4.0及以上版本实例查询。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/logical-nodes

表 5-285 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

无

响应参数

状态码：200

表 5-286 响应 Body 参数

参数	参数类型	描述
instance_id	String	实例ID。
total	Integer	节点数量。
nodes	Array of NodesInfoResp objects	节点信息。

表 5-287 NodesInfoResp

参数	参数类型	描述
logical_node_id	String	参数解释： 逻辑节点ID。 取值范围： 不涉及。
name	String	参数解释： 节点名称。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 节点状态。 取值范围： • Creating：创建中。 • Active：运行中。 • Inactive：故障。 • Deleting：删除中。 • AddSharding：添加分片中。
az_code	String	参数解释： 可用区code。 取值范围： 不涉及。
node_role	String	参数解释： 节点角色。 取值范围： • redis-server：Redis server节点。 • redis-proxy：proxy节点。
node_type	String	参数解释： 节点主从角色。 取值范围： • master：主节点。 • slave：从节点。 • proxy：proxy实例节点角色为"proxy"。
node_ip	String	参数解释： 节点的IP。 取值范围： 不涉及。
node_port	String	参数解释： 节点的端口。 取值范围： 不涉及。
node_id	String	参数解释： 节点ID。 取值范围： 不涉及。

参数	参数类型	描述
priority_weight	Integer	参数解释： 节点主备切换优先级权重。 取值范围： 0-100（ 1-100优先级逐步降低，1为最高，100为最低，0为禁止倒换 ）。
is_access	Boolean	参数解释： 节点的IP是否可直接访问。 取值范围： • true • false
group_id	String	参数解释： 实例分片ID。 取值范围： 不涉及。
group_name	String	参数解释： 实例分片名称。 取值范围： 不涉及。
is_remove_ip	Boolean	参数解释： 是否从只读域名中摘除IP。 取值范围： • true • false
replication_id	String	参数解释： 实例副本ID。 取值范围： 不涉及。

参数	参数类型	描述
dimensions	Array of InstanceReplicationDimensionsInfo objects	参数解释： 副本对应的监控指标维度信息。可用于调用云监控服务的查询监控数据指标相关接口。 • 副本的监控维度为多维度，返回数组中包含两个维度信息。从云监控查询监控数据时，要按多维度传递指标维度参数，才能查询到监控指标值。 • 第一个维度为副本父维度信息。维度名称为dcs_instance_id，维度值对应副本所在的实例ID。 • 第二个维度，维度名称为dcs_cluster_redis_node，维度值为副本的监控对象ID，与副本ID和节点ID不同。 取值范围： 不涉及。

表 5-288 InstanceReplicationDimensionsInfo

参数	参数类型	描述
name	String	参数解释： 监控维度名称。 取值范围： • dcs_instance_id：实例维度。 • dcs_cluster_redis_node：数据节点维度。
value	String	参数解释： 维度取值。 取值范围： 不涉及。

状态码：500

表 5-289 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：400

表 5-290 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/logical-nodes?status={status}
```


响应示例

状态码：200

查询节点信息成功。

```
{
  "instance_id": "11368fbb-de1e-4613-9080-1c374b71a1af",
  "total": 1,
  "nodes": [ {
    "logical_node_id": "79653bcbd0e5c104fefedd188aa372d6",
    "name": "(master)192.168.68.157",
    "status": "Active",
    "az_code": "region01",
    "node_role": "redis-server",
    "node_type": "master",
    "node_ip": "192.168.68.157",
    "node_id": "ff80808189bf92a50189d330303b77aa",
    "is_access": true,
    "group_id": "6e0f6e95-57cd-4baf-86d6-bea8815b25a8",
    "group_name": "group-0",
    "priority_weight": 100,
    "is_remove_ip": false,
    "dimensions": [ {
      "name": "dcs_instance_id",
      "value": "11368fbb-de1e-4613-9080-1c374b71a1af"
    }, {
      "name": "dcs_cluster_redis_node",
      "value": "79653bcbd0e5c104fefedd188aa372d6"
    } ],
    "replication_id": null
  } ]
}
```

状态码：500

内部服务错误。

```
{
  "error_code": "DCS.5000",
  "error_msg": "Internal service error."
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4800",
  "error_msg": "Invalid offset in the request."
}
```

状态码

状态码	描述
200	查询节点信息成功。
500	内部服务错误。
400	非法请求。

错误码

请参见[错误码](#)。

5.3.6 获取副本状态

功能介绍

获取分片副本状态。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instance/{instance_id}/groups/{group_id}/group-nodes-state

表 5-291 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
group_id	是	String	分片ID。

请求参数

无

响应参数

状态码：200

表 5-292 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of InstanceReplicationListInfo objects	副本列表

表 5-293 InstanceReplicationListInfo

参数	参数类型	描述
replication_role	String	参数解释： 副本角色。 取值范围： • master：表示主节点。 • slave：表示从节点。
replication_ip	String	参数解释： 副本IP。 取值范围： 不涉及。
is_replication	Boolean	参数解释： 是否为新增副本。 取值范围： • true • false
replication_id	String	参数解释： 副本ID。 取值范围： 不涉及。
node_id	String	参数解释： 节点ID。 取值范围： 不涉及。
status	String	参数解释： 副本状态。 取值范围： 不涉及。
az_code	String	参数解释： 副本所在的可用区。 取值范围： 不涉及。

参数	参数类型	描述
dimensions	Array of InstanceReplicationDimensionsInfo objects	参数解释： 副本对应的监控指标维度信息。可用于调用云监控服务的查询监控数据指标相关接口。 • 副本的监控维度为多维度，返回数组中包含两个维度信息。从云监控查询监控数据时，要按多维度传递指标维度参数，才能查询到监控指标值。 • 第一个维度为副本父维度信息，维度名称为dcs_instance_id，维度值对应副本所在的实例ID。 • 第二个维度，维度名称为dcs_cluster_redis_node，维度值为副本的监控对象ID，与副本ID和节点ID不同。 取值范围： 不涉及。

表 5-294 InstanceReplicationDimensionsInfo

参数	参数类型	描述
name	String	参数解释： 监控维度名称。 取值范围： • dcs_instance_id：实例维度。 • dcs_cluster_redis_node：数据节点维度。
value	String	参数解释： 维度取值。 取值范围： 不涉及。

状态码：500

表 5-295 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：400

表 5-296 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/groups/{group_id}/group-nodes-state
```

响应示例

状态码：200

查询分片和副本信息成功。

```
[ {
  "az_code" : "region01",
  "is_replication" : false,
  "node_id" : "8abfa7aa73291f15017329e1cda0002d",
  "replication_id" : "dfbd3f84-08bc-42f0-b538-01d03e6dc178",
  "replication_ip" : "192.168.76.25",
  "replication_role" : "master",
  "status" : "Active",
  "dimensions" : [ {
    "name" : "dcs_instance_id",
    "value" : "caf2d19f-7783-44b0-be46-8c9da3ef1e94"
  } ]
}]
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowReplicationStatesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowReplicationStatesRequest request = new ShowReplicationStatesRequest();
        request.withInstanceId("{instance_id}");
        request.withGroupId("{group_id}");
        try {
            ShowReplicationStatesResponse response = client.showReplicationStates(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowReplicationStatesRequest()
        request.instance_id = "{instance_id}"
        request.group_id = "{group_id}"
        response = client.show_replication_states(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowReplicationStatesRequest{}
request.InstanceId = "{instance_id}"
request.GroupId = "{group_id}"
response, err := client.ShowReplicationStates(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询分片和副本信息成功。
500	内部服务错误。
400	非法请求。

错误码

请参见[错误码](#)。

5.4 参数管理

5.4.1 查询实例配置参数

功能介绍

查询指定实例的配置参数信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/configs

表 5-297 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-298 响应 Body 参数

参数	参数类型	描述
config_time	String	参数解释： 实例操作时间。格式为： 2017-03-31T12:24:46.297Z 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
redis_config	Array of QueryRedisConfig objects	参数解释： 实例配置项数组。 取值范围： 不涉及。
config_status	String	参数解释： 实例修改状态。 取值范围： • UPDATING：更新中。 • FAILURE：修改失败。 • SUCCESS：修改成功。

参数	参数类型	描述
status	String	参数解释： 实例运行状态。 取值范围： • PAYING：订单未支付或处理中。 • CREATING：缓存实例开始创建。 • RUNNING：缓存实例运行中。 • CLOSING：缓存实例关闭中。 • STARTING：缓存实例启动中。 • RESTARTING：缓存实例重启中。 • CLOSED：缓存实例已经关闭。 • DELETING：缓存实例删除中。 • DELETED：缓存实例已删除。 • ERROR：缓存实例运行异常。 • FAILURE：缓存实例配置失败。 • SUCCESS：缓存实例配置成功。 • CREATEFAILED：缓存实例创建异常。 • DELETEFAILED：缓存实例删除异常。 • FREEZING：实例冻结。 • EXTENDING：缓存实例扩容中。 • EXTENDEDFAILED：实例扩容失败。 • MIGRATING：缓存实例迁移中。 • OTHER：其他状态。 • FLUSHING：实例清空数据中。 • RESTORING：备份数据恢复中。 • UPGRADING：实例软件升级中。 • IPEXCHANGING：IP交换中。 • IPROLLBACKING：IP回滚中。 • MS_SWAPPING：主备切换中。 • AZMigrating：AZ迁移中。

表 5-299 QueryRedisConfig

参数	参数类型	描述
param_value	String	参数解释： 配置参数值。 取值范围： 不涉及。
value_type	String	参数解释： 配置参数的值类型，例如：Integer，String。 取值范围： 不涉及。
value_range	String	参数解释： 配置参数的取值范围。 取值范围： 不涉及。
description	String	参数解释： 配置项的描述。 取值范围： 不涉及。
default_value	String	参数解释： 配置参数的默认值。 取值范围： 不涉及。
param_name	String	参数解释： 配置参数名称。 取值范围： 不涉及。
param_id	String	参数解释： 配置参数ID。 取值范围： 不涉及。

状态码：400

表 5-300 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-301 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/configs
```

响应示例

状态码：200

查询实例配置参数成功。

```
{
  "instance_id": "50829e5a-a4fc-4c01-8651-80be8a491602",
  "config_status": "SUCCESS",
  "config_time": "2020-07-06T07:04:31.464Z",
  "redis_config": [ {
    "param_id": "1",
    "param_name": "timeout",
    "description": "Close the connection after a client is idle for N seconds (0 to disable)",
    "param_value": "101",
    "value_range": "0-7200",
    "value_type": "Interger",
    "default_value": "0"
  } ],
  "status": "RUNNING"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListConfigurationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConfigurationsRequest request = new ListConfigurationsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListConfigurationsResponse response = client.listConfigurations(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```

        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListConfigurationsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_configurations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListConfigurationsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListConfigurations(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例配置参数成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.4.2 修改实例配置参数

功能介绍

为了确保分布式缓存服务发挥出最优性能，您可以根据自己的业务情况对DCS缓存实例的运行参数进行调整。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/configs

表 5-302 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-303 请求 Body 参数

参数	是否必选	参数类型	描述
redis_config	否	Array of RedisConfig objects	实例配置项数组。

表 5-304 RedisConfig

参数	是否必选	参数类型	描述
param_value	是	String	实例配置项的值。
param_name	是	String	实例配置项名。
param_id	是	String	实例配置项ID。

响应参数

状态码：204

修改实例参数成功。

状态码：400

表 5-305 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-306 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

修改实例配置参数，例如将timeout参数（客户端与服务空闲超时断开的的时间）设置为1000秒。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/configs

{
  "redis_config": [ {
    "param_id": "1",
    "param_name": "timeout",
    "param_value": "1000"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

修改实例配置参数，例如将timeout参数（客户端与服务空闲超时断开的的时间）设置为1000秒。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateConfigurationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateConfigurationsRequest request = new UpdateConfigurationsRequest();
        request.withInstanceId("{instance_id}");
        ModifyRedisConfigBody body = new ModifyRedisConfigBody();
        List<RedisConfig> listbodyRedisConfig = new ArrayList<>();
        listbodyRedisConfig.add(
            new RedisConfig()
                .withParamValue("1000")
                .withParamName("timeout")
        );
    }
}
```

```
        .withParamId("1")
    );
    body.withRedisConfig(listbodyRedisConfig);
    request.withBody(body);
    try {
        UpdateConfigurationsResponse response = client.updateConfigurations(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

修改实例配置参数，例如将timeout参数（客户端与服务空闲超时断开的时间）设置为1000秒。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateConfigurationsRequest()
        request.instance_id = "{instance_id}"
        listRedisConfigbody = [
            RedisConfig(
                param_value="1000",
                param_name="timeout",
                param_id="1"
            )
        ]
        request.body = ModifyRedisConfigBody(
            redis_config=listRedisConfigbody
        )
        response = client.update_configurations(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

Go

修改实例配置参数，例如将timeout参数（客户端与服务空闲超时断开的的时间）设置为1000秒。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateConfigurationsRequest{}
    request.InstanceId = "{instance_id}"
    var listRedisConfigbody = []model.RedisConfig{
        {
            ParamValue: "1000",
            ParamName: "timeout",
            ParamId: "1",
        },
    }
    request.Body = &model.ModifyRedisConfigBody{
        RedisConfig: &listRedisConfigbody,
    }
    response, err := client.UpdateConfigurations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	修改实例参数成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.4.3 异步修改实例配置参数

功能介绍

为了确保分布式缓存服务发挥出最优性能，您可以根据自己的业务情况对DCS缓存实例的运行参数进行调整。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/async-configs

表 5-307 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-308 请求 Body 参数

参数	是否必选	参数类型	描述
redis_config	否	Array of RedisConfig objects	实例配置项数组。

表 5-309 RedisConfig

参数	是否必选	参数类型	描述
param_value	是	String	实例配置项的值。
param_name	是	String	实例配置项名。
param_id	是	String	实例配置项ID。

响应参数

状态码：200

表 5-310 响应 Body 参数

参数	参数类型	描述
job_id	String	更新配置任务ID。

状态码：204

无实际修改参数，无异步任务，操作结束。

状态码：400

表 5-311 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：500

表 5-312 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/async-configs
{
  "redis_config": [{
    "param_id": "1",
    "param_name": "timeout",
    "param_value": "1000"
  }]
}
```

响应示例

状态码：200

发送异步修改参数成功。

```
{
  "job_id": "8abf6e1e7bceacf6017bcf3d9e880188"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
```



```
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.UUID;
import java.util.List;
import java.util.ArrayList;

public class UpdateInstanceConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceConfigRequest request = new UpdateInstanceConfigRequest();
        request.withInstanceId(UUID.fromString("{instance_id}"));
        ModifyRedisConfigBody body = new ModifyRedisConfigBody();
        List<RedisConfig> listbodyRedisConfig = new ArrayList<>();
        listbodyRedisConfig.add(
            new RedisConfig()
                .withParamValue("1000")
                .withParamName("timeout")
                .withParamId("1")
        );
        body.withRedisConfig(listbodyRedisConfig);
        request.withBody(body);
        try {
            UpdateInstanceConfigResponse response = client.updateInstanceConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateInstanceConfigRequest()
    request.instance_id = "{instance_id}"
    listRedisConfigbody = [
        RedisConfig(
            param_value="1000",
            param_name="timeout",
            param_id="1"
        )
    ]
    request.body = ModifyRedisConfigBody(
        redis_config=listRedisConfigbody
    )
    response = client.update_instance_config(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.UpdateInstanceConfigRequest{}
request.InstanceId = "{instance_id}"
var listRedisConfigbody = []model.RedisConfig{
    {
        ParamValue: "1000",
        ParamName: "timeout",
        ParamId: "1",
    },
}
request.Body = &model.ModifyRedisConfigBody{
    RedisConfig: &listRedisConfigbody,
}
response, err := client.UpdateInstanceConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	发送异步修改参数成功。
204	无实际修改参数，无异步任务，操作结束。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.4.4 查询实例参数修改记录列表

功能介绍

查询实例的参数修改记录列表，支持按照关键字查询。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/config-histories

表 5-313 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-314 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 每页显示条数。 约束限制： 不涉及。 取值范围： 1~1000 默认取值： 10

请求参数

无

响应参数

状态码：200

表 5-315 响应 Body 参数

参数	参数类型	描述
history_num	Integer	参数解释： 实例参数修改记录个数。 取值范围： 不涉及。
histories	Array of HistoryInfo objects	参数解释： 实例参数修改记录详情。

表 5-316 HistoryInfo

参数	参数类型	描述
history_id	String	参数解释： 修改记录ID。 取值范围： 不涉及。
type	String	参数解释： 修改类型。 取值范围： config_param：实例配置参数。

参数	参数类型	描述
created_at	String	参数解释： 修改时间。 取值范围： 不涉及。
status	String	参数解释： 修改状态。 取值范围： • SUCCESS: 实例参数修改成功。 • FAILURE: 实例参数修改失败。 • EXECUTING: 实例参数修改中。

状态码：400

表 5-317 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-318 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/config-histories?offset=0&limit=10
```

响应示例

状态码：200

参数解释：

查询实例参数修改记录列表成功。

```
{
  "history_num" : 1,
  "histories" : [ {
    "history_id" : "4ae8507f-7992-40e2-9928-41ccf1db4bdc",
    "type" : "config_param",
    "created_at" : "2022-10-20T03:37:44.636Z",
    "status" : "SUCCESS"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListConfigHistoriesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        ListConfigHistoriesRequest request = new ListConfigHistoriesRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListConfigHistoriesResponse response = client.listConfigHistories(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
```



```
.with_region(DcsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ListConfigHistoriesRequest()
    request.instance_id = "{instance_id}"
    response = client.list_config_histories(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListConfigHistoriesRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListConfigHistories(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 查询实例参数修改记录列表成功。
400	参数解释： 非法请求。
500	参数解释： 内部服务错误。

错误码

请参见[错误码](#)。

5.5 备份与恢复

5.5.1 删除备份文件

功能介绍

删除缓存实例已备份的文件。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/backups/{backup_id}

表 5-319 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份记录ID。
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

无

响应参数

状态码：204

删除备份文件成功。

状态码：400

表 5-320 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-321 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/backups/{backup_id}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteBackupFileSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBackupFileRequest request = new DeleteBackupFileRequest();
        request.withBackupId("{backup_id}");
        request.withInstanceId("{instance_id}");
        try {
            DeleteBackupFileResponse response = client.deleteBackupFile(request);
        } catch (Exception e) {
            // Handle exceptions
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBackupFileRequest()
        request.backup_id = "{backup_id}"
        request.instance_id = "{instance_id}"
        response = client.delete_backup_file(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteBackupFileRequest{}
request.BackupId = "{backup_id}"
request.InstanceId = "{instance_id}"
response, err := client.DeleteBackupFile(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除备份文件成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.5.2 恢复指定实例

功能介绍

恢复指定的缓存实例。

 说明

只有主备和集群类型的缓存实例支持备份恢复操作，单机实例不支持备份恢复操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/restores

表 5-322 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-323 请求 Body 参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份记录ID。
remark	否	String	恢复缓存实例的备注信息。

响应参数

状态码：200

表 5-324 响应 Body 参数

参数	参数类型	描述
restore_id	String	恢复记录ID。

状态码：400

表 5-325 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-326 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

恢复缓存实例的备份文件。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/restores
{
  "backup_id" : "8ba256cb-e5ac-44f6-a3da-c03d8f0e5029",
  "remark" : "restore instance"
}
```


响应示例

状态码：200

恢复指定实例成功。

```
{
  "restore_id" : "a6155972-800c-4170-a479-3231e907d2f6"
}
```

SDK 代码示例

SDK代码示例如下。

Java

恢复缓存实例的备份文件。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class RestoreInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        RestoreInstanceRequest request = new RestoreInstanceRequest();
        request.withInstanceId("{instance_id}");
        RestoreInstanceBody body = new RestoreInstanceBody();
        body.withRemark("restore instance");
        body.withBackupId("8ba256cb-e5ac-44f6-a3da-c03d8f0e5029");
        request.withBody(body);
        try {
            RestoreInstanceResponse response = client.restoreInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

恢复缓存实例的备份文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RestoreInstanceRequest()
        request.instance_id = "{instance_id}"
        request.body = RestoreInstanceBody(
            remark="restore instance",
            backup_id="8ba256cb-e5ac-44f6-a3da-c03d8f0e5029"
        )
        response = client.restore_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

恢复缓存实例的备份文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.RestoreInstanceRequest{}
request.InstanceId = "{instance_id}"
remarkRestoreInstanceBody:= "restore instance"
request.Body = &model.RestoreInstanceBody{
    Remark: &remarkRestoreInstanceBody,
    BackupId: "8ba256cb-e5ac-44f6-a3da-c03d8f0e5029",
}
response, err := client.RestoreInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	恢复指定实例成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.5.3 查询实例恢复记录

功能介绍

查询指定缓存实例的恢复记录列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/restores

表 5-327 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-328 Query 参数

参数	是否必选	参数类型	描述
begin_time	否	String	参数解释： 查询开始时间。 约束限制： 格式要求： yyyyMMddHHmmss，如： 20170718235959。 取值范围： 不涉及。 默认取值： 不涉及。
end_time	否	String	参数解释： 查询结束时间。 约束限制： 格式要求： yyyyMMddHHmmss，如： 20170718235959。 取值范围： 不涉及。 默认取值： 不涉及。
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示条数。 约束限制： 不涉及。 取值范围： 1~1000 默认取值： 10

请求参数

无

响应参数

状态码：200

表 5-329 响应 Body 参数

参数	参数类型	描述
restore_record_response	Array of InstanceRestoreInfo objects	参数解释： 实例恢复记录的详情数组。
total_num	Integer	参数解释： 实例恢复记录总数量。 取值范围： 不涉及。

表 5-330 InstanceRestoreInfo

参数	参数类型	描述
backup_id	String	参数解释： 备份记录ID。 取值范围： 不涉及。
restore_id	String	参数解释： 恢复记录ID。 取值范围： 不涉及。
backup_name	String	参数解释： 备份记录名称。 取值范围： 不涉及。
updated_at	String	参数解释： 恢复完成时间。 取值范围： 不涉及。

参数	参数类型	描述
restore_remark	String	参数解释： 恢复备注信息。 取值范围： 不涉及。
created_at	String	参数解释： 恢复任务创建时间。 取值范围： 不涉及。
progress	String	参数解释： 恢复进度。 取值范围： 0-100
error_code	String	参数解释： 恢复失败后错误码。 取值范围： • "dcs.08.0001": 启动备份恢复工具失败。 • "dcs.08.0002": 执行超时。 • "dcs.08.0003": 删除桶失败。 • "dcs.08.0004": 获取AK/SK失败。 • "dcs.08.0005": 创建桶失败。 • "dcs.08.0006": 查询备份数据大小失败。 • "dcs.08.0007": 恢复时同步数据失败。 • "dcs.08.0008": 自动备份任务未运行，实例正在运行其他任务。
restore_name	String	参数解释： 恢复记录名称。 取值范围： 不涉及。
backup_remark	String	参数解释： 备份备注信息。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 恢复状态。 取值范围： - waiting：等待中。 - restoring：恢复中。 - succeed：恢复成功。 - failed：恢复失败。

状态码：204
参数解释：
查询实例恢复当前无记录。

状态码：400

表 5-331 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-332 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/restores?offset={offset}&limit={limit}&begin_Time={begin_Time}&end_Time={end_Time}
```

响应示例

状态码：200

参数解释：

查询实例恢复记录成功。

```
{
  "restore_record_response": [ {
    "backup_id": "f4823e9e-fe9b-4ffd-be79-4e5d6de272bb",
    "restore_id": "a6155972-800c-4170-a479-3231e907d2f6",
    "backup_name": "backup_20170718000002",
    "updated_at": "2017-07-18T21:41:35.182Z",
    "restore_remark": "doctest",
    "created_at": "2017-07-18T21:41:20.721Z",
    "progress": "100.00",
    "error_code": { },
    "restore_name": "restore_20170718214120",
    "backup_remark": { },
    "status": "succeed"
  } ],
  "total_num": 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListRestoreRecordsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListRestoreRecordsRequest request = new ListRestoreRecordsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListRestoreRecordsResponse response = client.listRestoreRecords(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListRestoreRecordsRequest()
    request.instance_id = "{instance_id}"
    response = client.list_restore_records(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRestoreRecordsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListRestoreRecords(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 查询实例恢复记录成功。
204	参数解释： 查询实例恢复当前无记录。
400	参数解释： 非法请求。
500	参数解释： 内部服务错误。

错误码

请参见[错误码](#)。

5.5.4 获取备份文件下载链接

功能介绍

获取指定实例的备份文件下载链接，下载备份文件。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/backups/{backup_id}/links

表 5-333 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
backup_id	是	String	参数解释： 备份记录ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-334 请求 Body 参数

参数	是否必选	参数类型	描述
expiration	是	Integer	参数解释： 设置URL的有效期。 约束限制： 必须在5分钟到24小时之内，单位为秒。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-335 响应 Body 参数

参数	参数类型	描述
file_path	String	参数解释： OBS桶内文件路径。 取值范围： 不涉及。
bucket_name	String	参数解释： OBS桶名。 取值范围： 不涉及。
links	Array of LinkItem objects	参数解释： 备份文件下链接集合，链接数最大为64个。

表 5-336 LinksItem

参数	参数类型	描述
file_name	String	参数解释： 备份文件名称。 取值范围： 不涉及。
link	String	参数解释： 备份文件下载链接地址。 取值范围： 不涉及。

状态码：400

表 5-337 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-338 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

获取指定实例的备份文件下载链接，下载备份文件，设置URL有效期为1800秒。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/backups/{backup_id}/links
{
  "expiration": 1800
}
```

响应示例

状态码：200

参数解释：

获取备份文件下载链接成功。

```
{
  "file_path": "42489641-23c4-4855-bc89-befc85e2b7f7/ddfe5f66-a965-43ff-aec7-f3b489dc071b/",
  "bucket_name": "bucket5da9cf3bfabc4cae9023695b934e5e2b",
  "links": [ {
    "file_name": "redis_192.168.63.250_6379_10923-16383_20190820211816.rdb",
    "link": "https://bucket5da9cf3bfabc4cae9023695b934e5e2b.{obs_endpoint}:443/42489641-23c4-4855-bc89-befc85e2b7f7/ddfe5f66-a965-43ff-aec7-f3b489dc071b/redis_192.168.63.250_6379_10923-16383_20190820211816.rdb?AWSAccessKeyId=VD8CEQNG8VMQODUAAM0D&Expires=1566308915&Signature=s3l%2BrLbo%2BFZw%2BUsjVere%2FOQdKEg%3D"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

获取指定实例的备份文件下载链接，下载备份文件，设置URL有效期为1800秒。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListBackupFileLinksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBackupFileLinksRequest request = new ListBackupFileLinksRequest();
        request.withInstanceId("{instance_id}");
        request.withBackupId("{backup_id}");
        DownloadBackupFilesReq body = new DownloadBackupFilesReq();
        body.withExpiration(1800);
        request.withBody(body);
        try {
            ListBackupFileLinksResponse response = client.listBackupFileLinks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

获取指定实例的备份文件下载链接，下载备份文件，设置URL有效期为1800秒。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBackupFileLinksRequest()
        request.instance_id = "{instance_id}"
        request.backup_id = "{backup_id}"
        request.body = DownloadBackupFilesReq(
            expiration=1800
        )
        response = client.list_backup_file_links(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

获取指定实例的备份文件下载链接，下载备份文件，设置URL有效期为1800秒。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.ListBackupFileLinksRequest{}
request.InstanceId = "{instance_id}"
request.BackupId = "{backup_id}"
request.Body = &model.DownloadBackupFilesReq{
    Expiration: int32(1800),
}
response, err := client.ListBackupFileLinks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 获取备份文件下载链接成功。
400	参数解释： 非法请求。
500	参数解释： 内部服务错误。

错误码

请参见[错误码](#)。

5.5.5 备份指定实例

功能介绍

备份指定的缓存实例。

 **说明**
只有主备和集群类型的缓存实例支持备份恢复操作，单机实例不支持备份恢复操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/backups

表 5-339 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-340 请求 Body 参数

参数	是否必选	参数类型	描述
remark	否	String	参数解释： 备份缓存实例的备注信息。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
backup_format	否	String	参数解释： 备份缓存实例的格式。 约束限制： 不涉及。 取值范围： • aof • rdb 默认取值： rdb

响应参数

状态码：200

表 5-341 响应 Body 参数

参数	参数类型	描述
backup_id	String	参数解释： 备份记录ID。 取值范围： 不涉及。

状态码：400

表 5-342 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-343 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

备份指定的缓存实例，格式为aof。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/backups
{
  "remark": "Backup instances",
  "backup_format": "aof"
}
```

响应示例

状态码：200

参数解释：

备份指定缓存实例成功。

```
{
  "backup_id" : "548ceeff-2cbb-47ab-9a1c-7b085a8c08d7"
}
```

SDK 代码示例

SDK代码示例如下。

Java

备份指定的缓存实例，格式为aof。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CopyInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CopyInstanceRequest request = new CopyInstanceRequest();
        request.withInstanceId("{instance_id}");
        BackupInstanceBody body = new BackupInstanceBody();
        body.withBackupFormat(BackupInstanceBody.BackupFormatEnum.fromValue("aof"));
        body.withRemark("Backup instances");
        request.withBody(body);
        try {
            CopyInstanceResponse response = client.copyInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

备份指定的缓存实例，格式为aof。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CopyInstanceRequest()
        request.instance_id = "{instance_id}"
        request.body = BackupInstanceBody(
            backup_format="aof",
            remark="Backup instances"
        )
        response = client.copy_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

备份指定的缓存实例，格式为aof。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```



```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CopyInstanceRequest{
    request.InstanceId = "{instance_id}"
    backupFormatBackupInstanceBody:= model.GetBackupInstanceBodyBackupFormatEnum().AOF
    remarkBackupInstanceBody:= "Backup instances"
    request.Body = &model.BackupInstanceBody{
        BackupFormat: &backupFormatBackupInstanceBody,
        Remark: &remarkBackupInstanceBody,
    }
    response, err := client.CopyInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 备份指定缓存实例成功。
400	参数解释： 非法请求。
500	参数解释： 内部服务错误。

错误码

请参见[错误码](#)。

5.5.6 查询实例备份信息

功能介绍

查询指定缓存实例的备份信息列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/backups

表 5-344 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-345 Query 参数

参数	是否必选	参数类型	描述
begin_time	否	String	参数解释： 查询开始时间。 约束限制： 格式：yyyyMMddHHmmss， 如：20170718235959。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
end_time	否	String	参数解释： 查询结束时间。 约束限制： 格式：yyyyMMddHHmmss， 如：20170718235959。 取值范围： 不涉及。 默认取值： 不涉及。
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示条数。 约束限制： 不涉及。 取值范围： 1~1000 默认取值： 10

请求参数

无

响应参数

状态码：200

表 5-346 响应 Body 参数

参数	参数类型	描述
total_num	Integer	参数解释： 实例备份返回记录数。 取值范围： 不涉及。
backup_record_response	Array of BackupRecordResponse objects	参数解释： 备份信息的详情数组。

表 5-347 BackupRecordResponse

参数	参数类型	描述
backup_id	String	参数解释： 备份记录ID。 取值范围： 不涉及。
period	String	参数解释： 备份执行时间段。 取值范围： 不涉及。
backup_name	String	参数解释： 备份记录名称。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
size	Long	参数解释： 备份文件大小（ Byte ）。 取值范围： 不涉及。

参数	参数类型	描述
backup_type	String	参数解释： 备份类型。 取值范围： • manual：表示备份类型为手动备份。 • auto：表示备份类型为自动备份。
created_at	String	参数解释： 备份任务创建时间。 取值范围： 不涉及。
updated_at	String	参数解释： 备份完成时间。 取值范围： 不涉及。
progress	String	参数解释： 备份进度。 取值范围： 不涉及。
error_code	String	参数解释： 备份失败后错误码。 取值范围： • "dcs.08.0001"：启动备份恢复工具失败。 • "dcs.08.0002"：执行超时。 • "dcs.08.0003"：删除桶失败。 • "dcs.08.0004"：获取AK/SK失败。 • "dcs.08.0005"：创建桶失败。 • "dcs.08.0006"：查询备份数据大小失败。 • "dcs.08.0007"：恢复时同步数据失败。 • "dcs.08.0008"：自动备份任务未运行，实例正在运行其他任务。
remark	String	参数解释： 备份缓存实例的备注信息。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 备份状态。 取值范围： • waiting：等待中。 • backuping：备份中。 • succeed：备份成功。 • failed：备份失败。 • expired：备份文件过期。 • deleted：已手动删除备份文件。
is_support_restore	String	参数解释： 是否可以恢复。 取值范围： • TRUE：可恢复。 • FALSE：不可恢复。
execution_at	String	参数解释： 执行时间。 取值范围： 不涉及。
backup_format	String	参数解释： 备份类型。 取值范围： • aof • rdb

状态码：204
查询缓存实例备份当前无记录。

状态码：400

表 5-348 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-349 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/backups?offset={offset}&limit={limit}&beginTime={begin_Time}&end_time={end_Time}
```

响应示例

状态码：200

参数解释：
查询缓存实例备份信息成功。

```
{
  "total_num" : 1,
  "backup_record_response" : [ {
    "period" : { },
    "backup_type" : "manual",
    "created_at" : "2019-05-10T08:31:16.166Z",
    "remark" : "001",
    "is_support_restore" : "TRUE",
    "backup_id" : "4631832a-14c6-45b0-a0b3-3abd8f591ad1",
    "backup_name" : "backup_20190510163116",
    "instance_id" : "5560df16-cebf-4473-95c4-d1b573c16e79",
    "size" : 880232,
    "updated_at" : "2019-05-10T08:32:30.546Z",
    "progress" : "100.00",
    "error_code" : { },
    "status" : "succeed",
    "execution_at" : "2019-05-11T08:31:16.166Z",
    "backup_format" : "aof"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListBackupRecordsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBackupRecordsRequest request = new ListBackupRecordsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListBackupRecordsResponse response = client.listBackupRecords(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```



```

        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBackupRecordsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_backup_records(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

```

```
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListBackupRecordsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListBackupRecords(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询缓存实例备份信息成功。
204	查询缓存实例备份当前无记录。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6 数据迁移

5.6.1 进行 IP 交换

功能介绍

数据增量迁移时进行源实例与目标实例的IP交换。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task/{task_id}/exchange-ip

表 5-350 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
task_id	是	String	数据迁移任务ID

请求参数

表 5-351 请求 Body 参数

参数	是否必选	参数类型	描述
exchanged_ip	是	Array of strings	数据迁移中待交换的IP地址。
is_exchange_domain	是	Boolean	是否交换域名。

响应参数

状态码：204

IP交换成功。

状态码：400

表 5-352 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-353 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-354 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-355 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-356 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/migration-task/{task_id}/exchange-ip
{
  "exchanged_ip" : [ ],
  "is_exchange_domain" : true
}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
import java.util.List;
import java.util.ArrayList;

public class ExchangeInstanceIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ExchangeInstanceIpRequest request = new ExchangeInstanceIpRequest();
        request.withTaskId("{task_id}");
        IpExchangeRequest body = new IpExchangeRequest();
        body.withIsExchangeDomain(true);
        request.withBody(body);
        try {
            ExchangeInstanceIpResponse response = client.exchangeInstanceIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ExchangeInstanceIpRequest()
    request.task_id = "{task_id}"
    request.body = IpExchangeRequest(
        is_exchange_domain=True
    )
    response = client.exchange_instance_ip(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExchangeInstanceIpRequest{}
    request.TaskId = "{task_id}"
    request.Body = &model.IpExchangeRequest{
        IsExchangeDomain: true,
    }
    response, err := client.ExchangeInstanceIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	IP交换成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.2 IP 交换回滚

功能介绍

IP交换回滚。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task/{task_id}/rollback-ip

表 5-357 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 数据迁移任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：204
IP交换回滚成功。

状态码：400

表 5-358 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-359 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-360 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-361 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-362 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/migration-task/{task_id}/rollback-ip
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4990",
  "error_msg" : "Not support exchange ip."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class RollbackExchangeInstanceIpSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
RollbackExchangeInstanceIpRequest request = new RollbackExchangeInstanceIpRequest();
request.withTaskId("{task_id}");
try {
    RollbackExchangeInstanceIpResponse response = client.rollbackExchangeInstanceIp(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RollbackExchangeInstanceIpRequest()
        request.task_id = "{task_id}"
        response = client.rollback_exchange_instance_ip(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RollbackExchangeInstanceIpRequest{}
    request.TaskId = "{task_id}"
    response, err := client.RollbackExchangeInstanceIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	IP交换回滚成功。
400	非法请求。
401	认证信息非法。

状态码	描述
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.3 创建备份导入任务

功能介绍

创建备份导入任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task

表 5-363 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-364 请求 Body 参数

参数	是否必选	参数类型	描述
task_name	是	String	备份导入任务名称。
description	否	String	备份导入任务描述。

参数	是否必选	参数类型	描述
migration_type	是	String	迁移任务类型。 取值范围： • backupfile_import: 表示备份文件导入。 • online_migration: 表示在线迁移。目前已不支持使用该接口创建在线迁移任务，请调用 CreateOnlineMigrationTask 创建在线迁移任务。
migration_method	是	String	迁移方式，包括全量迁移和增量迁移两种类型。 取值范围： • full_amount_migration: 表示全量迁移。 • incremental_migration: 表示增量迁移。
backup_files	否	BackupFilesBody object	迁移任务类型为备份文件导入场景时的备份文件列表。
network_type	否	String	迁移任务类型为在线迁移时，表示源Redis和目标Redis连通的网络类型，包括vpc和vpn两种类型。目前已不支持用该接口创建在线迁移任务，请调用 CreateOnlineMigrationTask 创建在线迁移任务。
source_instance	否	SourceInstanceBody object	迁移任务类型为在线迁移场景时该参数为必填参数，表示源Redis信息。目前已不支持用该接口创建在线迁移任务，请调用 CreateOnlineMigrationTask 创建在线迁移任务。
target_instance	是	TargetInstanceBody object	表示目标Redis信息。

表 5-365 BackupFilesBody

参数	是否必选	参数类型	描述
file_source	是	String	数据来源，包含OBS桶和备份记录两种类型。
bucket_name	否	String	OBS桶名，如果数据来源是OBS桶的话，该参数为必填参数。

参数	是否必选	参数类型	描述
files	是	Array of Files objects	导入的备份文件列表，数据来源为OBS桶时必须填写。
backup_id	否	String	备份记录ID，数据来源为备份记录时必须填写。

表 5-366 Files

参数	是否必选	参数类型	描述
file_name	是	String	备份文件名。
size	否	String	文件大小（单位：Byte）。
update_at	否	String	文件最后修改时间（格式YYYY-MM-DD HH:MM:SS）。

表 5-367 SourceInstanceBody

参数	是否必选	参数类型	描述
addrs	是	String	Redis实例地址（source_instance信息中填写）。
password	否	String	Redis密码，如果设置了密码，则必须填写。
task_status	否	String	任务状态。
id	否	String	Redis实例ID。
ip	否	String	Redis IP地址。
port	否	String	Redis端口。
name	否	String	Redis名称。
proxy_multi_db	否	Boolean	proxy实例是否开启了多DB。
db	否	String	Redis数据库。

表 5-368 TargetInstanceBody

参数	是否必选	参数类型	描述
id	是	String	Redis实例ID（target_instance信息中必须填写）。

参数	是否必选	参数类型	描述
name	否	String	Redis实例名称（ target_instance信息中填写 ）。
password	否	String	Redis密码，如果设置了密码，则必须填写。
task_status	否	String	任务状态。
ip	否	String	Redis IP地址。
port	否	String	Redis端口。
addrs	否	String	Redis实例地址。
proxy_multi_db	否	Boolean	proxy实例是否开启了多DB。
db	否	String	Redis数据库。

响应参数

状态码：200

表 5-369 响应 Body 参数

参数	参数类型	描述
id	String	迁移任务ID。
name	String	迁移任务名称。
status	String	迁移任务状态，这个字段的值包括：SUCCESS（成功），FAILED（失败），MIGRATING（迁移中），TERMINATED（已结束）。
error	String	错误信息。

状态码：400

表 5-370 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-371 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-372 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-373 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-374 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

创建备份导入迁移任务，备份数据来源于OBS桶。

```
POST https://{dcs_endpoint}/v2/{project_id}/migration-task
{
  "task_name": "migration1",
  "description": "Test",
  "migration_type": "backupfile_import",
  "migration_method": "full_amount_migration",
  "backup_files": {
    "bucket_name": "bucketName",
    "file_source": "self_build_obs",
    "files": [ {
      "size": "754.0MB",
      "file_name": "fileName"
    } ]
  },
  "target_instance": {
    "id": "d2c8399f-bdba-43c1-8069-3ff17a1a79d5",
    "name": "dcs-test",
    "password": "xxxxxx"
  }
}
```

响应示例

状态码：200

创建备份导入任务成功。

```
{
  "id": "8aa6999e71cb638b0171f485f5266ef0",
  "name": "dcs-test",
  "status": "MIGRATING",
}
```

```
"error": ""  
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建备份导入迁移任务，备份数据来源于OBS桶。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateMigrationTaskSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateMigrationTaskRequest request = new CreateMigrationTaskRequest();  
        CreateMigrationTaskBody body = new CreateMigrationTaskBody();  
        TargetInstanceBody targetInstancebody = new TargetInstanceBody();  
        targetInstancebody.withId("d2c8399f-bdba-43c1-8069-3ff17a1a79d5")  
            .withName("dcs-test")  
            .withPassword("xxxxxx");  
        List<Files> listBackupFilesFiles = new ArrayList<>();  
        listBackupFilesFiles.add(  
            new Files()  
                .withFileName("fileName")  
                .withSize("754.0MB")  
        );  
        BackupFilesBody backupFilesbody = new BackupFilesBody();  
        backupFilesbody.withFileSource(BackupFilesBody.FileSourceEnum.fromValue("self_build_obs"))  
            .withBucketName("bucketName")  
            .withFiles(listBackupFilesFiles);  
        body.withTargetInstance(targetInstancebody);  
        body.withBackupFiles(backupFilesbody);  
  
        body.withMigrationMethod(CreateMigrationTaskBody.MigrationMethodEnum.fromValue("full_amount_migra-  
tion"));
```

```
body.withMigrationType(CreateMigrationTaskBody.MigrationTypeEnum.fromValue("backupfile_import"));
body.withDescription("Test");
body.withTaskName("migration1");
request.withBody(body);
try {
    CreateMigrationTaskResponse response = client.createMigrationTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

创建备份导入迁移任务，备份数据来源于OBS桶。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateMigrationTaskRequest()
        targetInstancebody = TargetInstanceBody(
            id="d2c8399f-bdba-43c1-8069-3ff17a1a79d5",
            name="dcs-test",
            password="xxxxxx"
        )
        listFilesBackupFiles = [
            Files(
                file_name="fileName",
                size="754.0MB"
            )
        ]
        backupFilesbody = BackupFilesBody(
            file_source="self_build_obs",
            bucket_name="bucketName",
            files=listFilesBackupFiles
        )
```



```
request.body = CreateMigrationTaskBody(  
    target_instance=targetInstancebody,  
    backup_files=backupFilesbody,  
    migration_method="full_amount_migration",  
    migration_type="backupfile_import",  
    description="Test",  
    task_name="migration1"  
)  
response = client.create_migration_task(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

创建备份导入迁移任务，备份数据来源于OBS桶。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dcs.NewDcsClient(  
        dcs.DcsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateMigrationTaskRequest{  
        nameTargetInstance:= "dcs-test"  
        passwordTargetInstance:= "xxxxxx"  
        targetInstancebody := &model.TargetInstanceBody{  
            Id: "d2c8399f-bdba-43c1-8069-3ff17a1a79d5",  
            Name: &nameTargetInstance,  
            Password: &passwordTargetInstance,  
        }  
        sizeFiles:= "754.0MB"  
        var listFilesBackupFiles = []model.Files{  
            {  
                FileName: "fileName",  
                Size: &sizeFiles,  
            },  
        }  
        fileSourceBackupFiles:= model.GetBackupFilesBodyFileSourceEnum().SELF_BUILD_OBS  
        bucketNameBackupFiles:= "bucketName"
```

```
backupFilesbody := &model.BackupFilesBody{
    FileSource: &fileSourceBackupFiles,
    BucketName: &bucketNameBackupFiles,
    Files: listFilesBackupFiles,
}
descriptionCreateMigrationTaskBody:= "Test"
request.Body = &model.CreateMigrationTaskBody{
    TargetInstance: targetInstancebody,
    BackupFiles: backupFilesbody,
    MigrationMethod:
model.GetCreateMigrationTaskBodyMigrationMethodEnum().FULL_AMOUNT_MIGRATION,
    MigrationType: model.GetCreateMigrationTaskBodyMigrationTypeEnum().BACKUPFILE_IMPORT,
    Description: &descriptionCreateMigrationTaskBody,
    TaskName: "migration1",
}
response, err := client.CreateMigrationTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建备份导入任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.4 查询迁移任务列表

功能介绍

查询迁移任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/migration-tasks

表 5-375 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-376 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10

参数	是否必选	参数类型	描述
name	否	String	参数解释： 迁移任务名称。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-377 响应 Body 参数

参数	参数类型	描述
count	Integer	参数解释： 迁移任务数量。 取值范围： 不涉及。
migration_tasks	Array of MigrationTaskList objects	参数解释： 迁移任务列表。 取值范围： 不涉及。

表 5-378 MigrationTaskList

参数	参数类型	描述
task_id	String	参数解释： 迁移任务ID。 取值范围： 不涉及。

参数	参数类型	描述
task_name	String	参数解释： 迁移任务名称。 取值范围： 不涉及。
status	String	参数解释： 迁移任务状态。 取值范围： • SUCCESS：成功。 • FAILED：失败。 • MIGRATING：迁移中。 • TERMINATED：已结束。
migration_type	String	参数解释： 迁移任务类型。 取值范围： • backupfile_import：备份文件导入。 • online_migration：在线迁移。
migration_method	String	参数解释： 迁移方式。 取值范围： • full_amount_migration：全量迁移。 • incremental_migration：增量迁移。
ecs_tenant_private_ip	String	参数解释： 迁移机租户侧私有IP，与目的/源redis私有IP处于同VPC，可将此IP加入白名单。 取值范围： 不涉及。
data_source	String	参数解释： 源redis地址，格式为ip:port或者桶名。 取值范围： 不涉及。
source_instance_name	String	参数解释： 源实例名称，若源实例为自建redis，该参数为空。 取值范围： 不涉及。

参数	参数类型	描述
source_instance_id	String	参数解释： 源实例ID，若源实例为自建redis，该参数为空。 取值范围： 不涉及。
target_instance_address	String	参数解释： 目标redis地址，格式为ip:port。 取值范围： 不涉及。
target_instance_name	String	参数解释： 目标实例名称。 取值范围： 不涉及。
target_instance_id	String	参数解释： 目标实例ID。 取值范围： 不涉及。
created_at	String	参数解释： 迁移任务创建时间。 取值范围： 不涉及。
description	String	参数解释： 迁移任务描述。 取值范围： 不涉及。
source_instance_status	String	参数解释： 源实例状态，若源实例为自建redis，该参数为空。 取值范围： 不涉及。
target_instance_status	String	参数解释： 目标实例状态。 取值范围： 不涉及。

参数	参数类型	描述
source_instance_subnet_id	String	参数解释： 源实例子网ID，若源实例为自建redis，该参数为空。 取值范围： 不涉及。
target_instance_subnet_id	String	参数解释： 目标实例子网ID。 取值范围： 不涉及。
source_instance_spec_code	String	参数解释： 源实例规格编码，若源实例为自建redis，该参数为空。 取值范围： 不涉及。
target_instance_spec_code	String	参数解释： 目标实例规格编码。 取值范围： 不涉及。
error_message	String	参数解释： 错误信息。 取值范围： 不涉及。
released_at	String	参数解释： 迁移机释放时间。 取值范围： 不涉及。
version	String	参数解释： 迁移任务版本。 取值范围： 不涉及。
resume_mode	String	参数解释： 迁移任务的重连模式。 取值范围： • auto：自动重连。 • manual：手动重连。

参数	参数类型	描述
supported_features	Array of strings	参数解释： 支持的特性。 取值范围： 不涉及。

状态码：400

表 5-379 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-380 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-381 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-382 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-383 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/666486c2d9b948c1bbea57e714d744fa/migration-tasks?offset=0&limit=10&name=dcs-migration-i9qk
```

响应示例

状态码：200

查询迁移任务列表。

```
{  
  "count" : 1,
```

```
"migration_tasks": [ {
  "task_id": "2944a563-6f27-4604-9c16-a9a53cd95cc7",
  "task_name": "dcs-migration-test-6-16",
  "status": "TERMINATED",
  "migration_type": "online_migration",
  "migration_method": "incremental_migration",
  "data_source": "192.168.0.200:6379",
  "source_instance_name": "migration-resource",
  "source_instance_id": "132e0017-715b-4556-930a-95c7db7e8f7a",
  "target_instance_addr": "192.168.0.153:6379",
  "target_instance_name": "migration-target",
  "target_instance_id": "eb1a8d1c-9ea6-4fcd-ac43-c3d3c2493f06",
  "created_at": "2021-06-16T06:12:55.508Z",
  "description": "dcs-test",
  "source_instance_status": "RUNNING",
  "target_instance_status": "RUNNING",
  "source_instance_subnet_id": "192.168.0.1",
  "target_instance_subnet_id": "192.168.0.2",
  "source_instance_spec_code": "redis.single.xu1.tiny.128",
  "target_instance_spec_code": "redis.single.xu1.tiny.128",
  "error_message": null,
  "released_at": "202404011057",
  "version": "6.0.21",
  "resume_mode": "auto",
  "supported_features": [ ]
} ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListMigrationTaskRequest request = new ListMigrationTaskRequest();
```

```
try {
    ListMigrationTaskResponse response = client.listMigrationTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListMigrationTaskRequest()
        response = client.list_migration_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListMigrationTaskRequest{}
response, err := client.ListMigrationTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询迁移任务列表。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.5 删除数据迁移任务

功能介绍

删除数据迁移任务。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/migration-tasks/delete

表 5-384 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-385 请求 Body 参数

参数	是否必选	参数类型	描述
task_id_list	是	Array of strings	参数解释： 删除的迁移任务ID列表。 约束限制： 不涉及。

响应参数

状态码：200

表 5-386 响应 Body 参数

参数	参数类型	描述
task_id_list	Array of strings	参数解释： 删除的迁移任务ID列表。

状态码：400

表 5-387 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-388 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-389 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-390 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-391 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

删除数据迁移任务。

```
DELETE https://{dcs_endpoint}/v2/666486c2d9b948c1bbea57e714d744fa/migration-tasks/delete
{
  "task_id_list" : [ "2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca" ]
}
```

响应示例

状态码：200

参数解释：

当迁移任务ID存在时，删除数据迁移任务成功；当迁移任务ID不存在时，将不做处理。

```
{
  "task_id_list" : [ "2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除数据迁移任务。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteMigrationTaskRequest request = new DeleteMigrationTaskRequest();
        DeleteMigrateTaskRequest body = new DeleteMigrateTaskRequest();
        List<String> listbodyTaskIdList = new ArrayList<>();
        listbodyTaskIdList.add("2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca");
        body.withTaskIdList(listbodyTaskIdList);
        request.withBody(body);
        try {
            DeleteMigrationTaskResponse response = client.deleteMigrationTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

删除数据迁移任务。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteMigrationTaskRequest()
    listTaskIdListbody = [
        "2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca"
    ]
    request.body = DeleteMigrateTaskRequest(
        task_id_list=listTaskIdListbody
    )
    response = client.delete_migration_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

删除数据迁移任务。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteMigrationTaskRequest{}
    var listTaskIdListbody = []string{
```

```
        "2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca",
    }
    request.Body = &model.DeleteMigrateTaskRequest{
        TaskIdList: listTaskIdListbody,
    }
    response, err := client.DeleteMigrationTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 当迁移任务ID存在时，删除数据迁移任务成功；当迁移任务ID不存在时，将不做处理。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在（无返回，见200状态码）。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.6 查询迁移任务详情

功能介绍

查询迁移任务详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/migration-task/{task_id}

表 5-392 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 数据迁移任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-393 响应 Body 参数

参数	参数类型	描述
task_id	String	参数解释： 迁移任务ID。 取值范围： 不涉及。
task_name	String	参数解释： 迁移任务名称。 取值范围： 不涉及。

参数	参数类型	描述
description	String	参数解释： 迁移任务描述。 取值范围： 不涉及。
status	String	参数解释： 迁移任务状态。 取值范围： • SUCCESS: 成功 • FAILED: 失败 • MIGRATING: 迁移中 • TERMINATED: 已结束 • TERMINATING: 停止中 • INCRMIGEATING: 增量迁移中 • FULLMIGRATING: 全量迁移中 • RUNNING: 待配置 • MIGRATION_FAILED: 迁移失败 • RELEASED: 已释放
migration_type	String	参数解释： 迁移任务描述。 取值范围： • backupfile_import: 备份导入 • online_migration: 在线迁移
migration_method	String	参数解释： 迁移方式。 取值范围： • full_amount_migration: 全量迁移。 • incremental_migration: 在线迁移。
ecs_tenant_private_ip	String	参数解释： 迁移机租户侧私有IP，与目的/源redis私有IP处于同VPC，可将此IP加入白名单。 取值范围： 不涉及。
backup_files	BackupFilesRespBody object	参数解释： 备份文件导入场景的备份文件列表。

参数	参数类型	描述
network_type	String	参数解释： 网络类型，包括vpc和vpn两种类型。 取值范围： - vpc - vpn
source_instance	SourceInstanceRespBody object	参数解释： 源Redis信息。
target_instance	TargetInstanceRespBody object	参数解释： 目标Redis信息。
created_at	String	参数解释： 迁移任务创建时间。 取值范围： 不涉及。
updated_at	String	参数解释： 迁移任务完成时间。 取值范围： 不涉及。
released_at	String	参数解释： 迁移机释放时间。 取值范围： 不涉及。
version	String	参数解释： 迁移机版本。 取值范围： 不涉及。
resume_mode	String	参数解释： 数据迁移操作模式。 取值范围： - auto：自动迁移。 - manual：手动迁移。
supported_features	Array of strings	参数解释： 支持的特性。 取值范围： 不涉及。

参数	参数类型	描述
tenant_vpc_id	String	参数解释： 租户VPC ID。 取值范围： 不涉及。
tenant_subnet_id	String	参数解释： 租户子网ID。 取值范围： 不涉及。
tenant_security_group_id	String	参数解释： 租户安全组ID。 取值范围： 不涉及。
bandwidth_limit_mb	String	参数解释： 带宽限制速度。 取值范围： 不涉及。
task_status	String	参数解释： 任务状态，只有在交换IP时会返回EXCHANGING。 取值范围： EXCHANGING: IP交换中。

表 5-394 BackupFilesRespBody

参数	参数类型	描述
file_source	String	参数解释： 数据来源，包含OBS桶和备份记录两种类型。 取值范围： - self_build_obs: obs桶数据。 - backup_record: Redis备份文件。
bucket_name	String	参数解释： OBS桶名。 取值范围： 不涉及。

参数	参数类型	描述
files	Array of FilesResp objects	参数解释： 导入的备份文件列表。
backup_id	String	参数解释： 备份记录ID。 取值范围： 不涉及。

表 5-395 FilesResp

参数	参数类型	描述
file_name	String	参数解释： 备份文件名。 取值范围： 不涉及。
size	String	参数解释： 文件大小（单位：Byte）。 取值范围： 不涉及。
update_at	String	参数解释： 文件最后修改时间（格式YYYY-MM-DD HH:MM:SS）。 取值范围： 不涉及。

表 5-396 SourceInstanceRespBody

参数	参数类型	描述
addrs	String	参数解释： Redis实例IP和域名地址。 取值范围： 不涉及。
password	String	参数解释： Redis密码。 取值范围： 不涉及。

参数	参数类型	描述
id	String	参数解释： Redis实例ID。 取值范围： 不涉及。
ip	String	参数解释： Redis IP地址。 取值范围： 不涉及。
port	String	参数解释： Redis端口。 取值范围： 不涉及。
name	String	参数解释： Redis名称。 取值范围： 不涉及。
proxy_multi_db	Boolean	参数解释： Proxy实例是否开启了多DB。 取值范围： • true：开启。 • false：未开启。
db	String	参数解释： Redis数据库。 取值范围： 0-255。
withProxy	Boolean	参数解释： 是否有Proxy节点。 取值范围： • true：带有Proxy节点。 • false：不带有Proxy节点。

表 5-397 TargetInstanceRespBody

参数	参数类型	描述
addr	String	参数解释： Redis实例地址（ source_instance信息中填写 ）。 取值范围： 不涉及。
password	String	参数解释： Redis密码。 取值范围： 不涉及。
id	String	参数解释： Redis实例ID。 取值范围： 不涉及。
ip	String	参数解释： Redis IP地址。 取值范围： 不涉及。
port	String	参数解释： Redis端口。 取值范围： 不涉及。
name	String	参数解释： Redis名称。 取值范围： 不涉及。
proxy_multi_db	Boolean	参数解释： Proxy实例是否开启了多DB。 取值范围： ● true： 开启。 ● false： 未开启。
db	String	参数解释： Redis数据库。 取值范围： 0-255。

参数	参数类型	描述
withProxy	Boolean	参数解释： 是否有Proxy节点。 取值范围： • true：带有Proxy节点。 • false：不带有Proxy节点。

状态码：400

表 5-398 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-399 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-400 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-401 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-402 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/254d2280acb74d36a8b839746bcd7352 /migration-task/a276c1ffa0a54c2597fc824b917db317
```

响应示例

状态码：200

参数解释：

查询迁移任务详情成功。

```
{
  "task_name" : "migration1",
  "task_id" : "a276c1ffa0a54c2597fc824b917db317",
  "status" : "TERMINATING",
  "migration_type" : "incremental_migration",
  "network_type" : "vpc",
  "source_instance" : {
    "addr" : "192.168.1.2:6379",
    "id" : "86157b69-8a62-47f9-8c55-fa6ddc8927e1",
    "name" : "dcs-test",
    "ip" : "192.168.1.2",
    "port" : "6379",
    "proxy_multi_db" : false,
    "password" : "DCStest123",
    "db" : "0"
  },
  "target_instance" : {
    "addr" : "192.168.1.2:6379",
    "id" : "86157b69-8a62-47f9-8c55-fa6ddc8927ee",
    "name" : "test-instance",
    "ip" : "192.168.1.2",
    "port" : "6379",
    "proxy_multi_db" : false,
    "password" : "DCStest123",
    "db" : "0"
  },
  "created_at" : "2019/10/22 14:44:30",
  "updated_at" : "2019/10/22 18:00:00"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowMigrationTaskRequest request = new ShowMigrationTaskRequest();
request.withTaskId("{task_id}");
try {
    ShowMigrationTaskResponse response = client.showMigrationTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMigrationTaskRequest()
        request.task_id = "{task_id}"
        response = client.show_migration_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMigrationTaskRequest{}
    request.TaskId = "{task_id}"
    response, err := client.ShowMigrationTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询迁移任务详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.7 设置迁移任务自动重连

功能介绍

设置迁移任务自动重连。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/migration-task/{task_id}

表 5-403 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 数据迁移任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-404 请求 Body 参数

参数	是否必选	参数类型	描述
resume_mode	否	String	参数解释： 迁移任务重连模式。 约束限制： 不涉及。 取值范围： • auto：自动重连。自动重连模式在遇到网络等异常情况时，会无限自动重试。自动重连模式在无法进行增量同步时，会触发全量同步，增加带宽占用，请谨慎选择。 • manual：手动重连。 默认取值： auto。

响应参数

状态码：200
查询后台任务详情成功。

状态码：400

表 5-405 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见错误码。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-406 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-407 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-408 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-409 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/migration-task/{task_id}

{
  "resume_mode" : "auto"
}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid task_id parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class UpdateMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateMigrationTaskRequest request = new UpdateMigrationTaskRequest();
        request.withTaskId("{task_id}");
        MigrationUpdateRequestEntity body = new MigrationUpdateRequestEntity();
        body.withResumeMode("auto");
        request.withBody(body);
        try {
            UpdateMigrationTaskResponse response = client.updateMigrationTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = UpdateMigrationTaskRequest()
    request.task_id = "{task_id}"
    request.body = MigrationUpdateRequestEntity(
        resume_mode="auto"
    )
    response = client.update_migration_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateMigrationTaskRequest{
        request.TaskId = "{task_id}"
        resumeModeMigrationUpdateRequestEntity:= "auto"
        request.Body = &model.MigrationUpdateRequestEntity{
            ResumeMode: &resumeModeMigrationUpdateRequestEntity,
        }
    }
    response, err := client.UpdateMigrationTask(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询后台任务详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.8 停止数据迁移任务

功能介绍

停止数据迁移任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task/{task_id}/stop

表 5-410 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
task_id	是	String	参数解释： 数据迁移任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-411 响应 Body 参数

参数	参数类型	描述
task_id	String	参数解释： 迁移任务ID。 取值范围： 不涉及。
task_name	String	参数解释： 迁移任务名称。 取值范围： 不涉及。
description	String	参数解释： 迁移任务描述。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 迁移任务状态。 取值范围： • SUCCESS：成功 • FAILED：失败 • MIGRATING：迁移中 • TERMINATED：已结束 • TERMINATING：停止中 • INCRMIGEATING：增量迁移中 • FULLMIGRATING：全量迁移中 • RUNNING：待配置 • MIGRATION_FAILED：迁移失败 • RELEASED：已释放
migration_type	String	参数解释： 迁移任务描述。 取值范围： • backupfile_import：备份导入 • online_migration：在线迁移
migration_method	String	参数解释： 迁移方式。 取值范围： • full_amount_migration：全量迁移。 • incremental_migration：在线迁移。
ecs_tenant_private_ip	String	参数解释： 迁移机租户侧私有IP，与目的/源redis私有IP处于同VPC，可将此IP加入白名单。 取值范围： 不涉及。
backup_files	BackupFilesResp Body object	参数解释： 备份文件导入场景的备份文件列表。
network_type	String	参数解释： 网络类型，包括vpc和vpn两种类型。 取值范围： • vpc • vpn

参数	参数类型	描述
source_instance	SourceInstanceRespBody object	参数解释： 源Redis信息。
target_instance	TargetInstanceRespBody object	参数解释： 目标Redis信息。
created_at	String	参数解释： 迁移任务创建时间。 取值范围： 不涉及。
updated_at	String	参数解释： 迁移任务完成时间。 取值范围： 不涉及。
released_at	String	参数解释： 迁移机释放时间。 取值范围： 不涉及。
version	String	参数解释： 迁移机版本。 取值范围： 不涉及。
resume_mode	String	参数解释： 数据迁移操作模式。 取值范围： • auto：自动迁移。 • manual：手动迁移。
supported_features	Array of strings	参数解释： 支持的特性。 取值范围： 不涉及。
tenant_vpc_id	String	参数解释： 租户VPC ID。 取值范围： 不涉及。

参数	参数类型	描述
tenant_subnet_id	String	参数解释： 租户子网ID。 取值范围： 不涉及。
tenant_security_group_id	String	参数解释： 租户安全组ID。 取值范围： 不涉及。
bandwidth_limit_mb	String	参数解释： 带宽限制速度。 取值范围： 不涉及。
task_status	String	参数解释： 任务状态，只有在交换IP时会返回EXCHANGING。 取值范围： EXCHANGING: IP交换中。

表 5-412 BackupFilesResponseBody

参数	参数类型	描述
file_source	String	参数解释： 数据来源，包含OBS桶和备份记录两种类型。 取值范围： - self_build_obs: obs桶数据。 - backup_record: Redis备份文件。
bucket_name	String	参数解释： OBS桶名。 取值范围： 不涉及。
files	Array of FilesResp objects	参数解释： 导入的备份文件列表。

参数	参数类型	描述
backup_id	String	参数解释： 备份记录ID。 取值范围： 不涉及。

表 5-413 FilesResp

参数	参数类型	描述
file_name	String	参数解释： 备份文件名。 取值范围： 不涉及。
size	String	参数解释： 文件大小（单位：Byte）。 取值范围： 不涉及。
update_at	String	参数解释： 文件最后修改时间（格式YYYY-MM-DD HH:MM:SS）。 取值范围： 不涉及。

表 5-414 SourceInstanceRespBody

参数	参数类型	描述
addrs	String	参数解释： Redis实例IP和域名地址。 取值范围： 不涉及。
password	String	参数解释： Redis密码。 取值范围： 不涉及。

参数	参数类型	描述
id	String	参数解释： Redis实例ID。 取值范围： 不涉及。
ip	String	参数解释： Redis IP地址。 取值范围： 不涉及。
port	String	参数解释： Redis端口。 取值范围： 不涉及。
name	String	参数解释： Redis名称。 取值范围： 不涉及。
proxy_multi_db	Boolean	参数解释： Proxy实例是否开启了多DB。 取值范围： • true：开启。 • false：未开启。
db	String	参数解释： Redis数据库。 取值范围： 0-255。
withProxy	Boolean	参数解释： 是否有Proxy节点。 取值范围： • true：带有Proxy节点。 • false：不带有Proxy节点。

表 5-415 TargetInstanceRespBody

参数	参数类型	描述
addr	String	参数解释： Redis实例地址（source_instance信息中填写）。 取值范围： 不涉及。
password	String	参数解释： Redis密码。 取值范围： 不涉及。
id	String	参数解释： Redis实例ID。 取值范围： 不涉及。
ip	String	参数解释： Redis IP地址。 取值范围： 不涉及。
port	String	参数解释： Redis端口。 取值范围： 不涉及。
name	String	参数解释： Redis名称。 取值范围： 不涉及。
proxy_multi_db	Boolean	参数解释： Proxy实例是否开启了多DB。 取值范围： • true：开启。 • false：未开启。
db	String	参数解释： Redis数据库。 取值范围： 0-255。

参数	参数类型	描述
withProxy	Boolean	参数解释： 是否有Proxy节点。 取值范围： • true：带有Proxy节点。 • false：不带有Proxy节点。

状态码：400

表 5-416 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-417 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-418 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-419 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-420 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/666486c2d9b948c1bbee57e714d744fa/migration-task/2fb6b7e2-5eb8-4380-9d60-ce8d12b19aca/stop
```

响应示例

状态码：200

停止数据迁移任务成功。

```
{
  "task_name": "migration1",
  "task_id": "a276c1ffa0a54c2597fc824b917db317",
  "status": "TERMINATING",
  "migration_type": "incremental_migration",
  "network_type": "vpc",
  "source_instance": {
    "addr": "192.168.1.2:6379",
    "id": "86157b69-8a62-47f9-8c55-fa6ddc8927e1",
    "name": "dcs-test",
    "ip": "192.168.1.2",
    "port": "6379",
    "proxy_multi_db": false,
    "password": "DCStest123",
    "db": "0"
  },
  "target_instance": {
    "addr": "192.168.1.2:6379",
    "id": "86157b69-8a62-47f9-8c55-fa6ddc8927ee",
    "name": "test-instance",
    "ip": "192.168.1.2",
    "port": "6379",
    "proxy_multi_db": false,
    "password": "DCStest123",
    "db": "0"
  },
  "created_at": "2019/10/22 14:44:30",
  "updated_at": "2019/10/22 18:00:00"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class StopMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
StopMigrationTaskRequest request = new StopMigrationTaskRequest();
request.withTaskId("{task_id}");
try {
    StopMigrationTaskResponse response = client.stopMigrationTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = StopMigrationTaskRequest()
        request.task_id = "{task_id}"
        response = client.stop_migration_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopMigrationTaskRequest{}
    request.TaskId = "{task_id}"
    response, err := client.StopMigrationTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	停止数据迁移任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.9 查询在线迁移进度明细

功能介绍

查询在线迁移进度明细。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/migration-task/{task_id}/stats

表 5-421 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
task_id	是	String	任务ID。

请求参数

无

响应参数

状态码：200

表 5-422 响应 Body 参数

参数	参数类型	描述
full_migration_progress	String	全量迁移进度百分比。
offset	String	增量迁移偏移量，是增量迁移阶段源redis和目标redis数据差值的总和，如果offset为0，则表示当前增量迁移阶段源redis和目标redis无偏差。
source_dbsize	String	源redis键数量。
target_dbsize	String	目标redis键数量。
target_input_kbps	String	目标redis键写入流量，单位KB/s。

参数	参数类型	描述
target_ops	String	目标redis每秒并发操作数。
is_migrating	Boolean	迁移任务是否在进行。

状态码：400

表 5-423 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-424 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-425 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-426 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-427 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
https://{dcs_endpoint}/v2/666486c2d9b948c1bbea57e714d744fa/migration-task/31c6fd2b-2893-430b-90ff-d1adb8060585/stats
```

响应示例

状态码：200

查询在线迁移进度明细成功。

```
{  
  "full_migration_progress" : 100,  
}
```

```
"offset" : 100,  
"source_dbsize" : 531221,  
"target_dbsize" : 531211,  
"target_input_kbps" : 50,  
"target_ops" : 5300,  
"is_migrating" : true  
}
```

状态码：400

非法请求。

```
{  
  "error_code" : "DCS.4133",  
  "error_msg" : "Instance job is not found."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
public class ShowMigrationTaskStatsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowMigrationTaskStatsRequest request = new ShowMigrationTaskStatsRequest();  
        request.withTaskId("{task_id}");  
        try {  
            ShowMigrationTaskStatsResponse response = client.showMigrationTaskStats(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
        }  
    }  
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMigrationTaskStatsRequest()
        request.task_id = "{task_id}"
        response = client.show_migration_task_stats(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```
WithSk(sk).
WithProjectId(projectId).
Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowMigrationTaskStatsRequest{}
request.TaskId = "{task_id}"
response, err := client.ShowMigrationTaskStats(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询在线迁移进度明细成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.10 创建在线数据迁移任务

功能介绍

创建在线数据迁移任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration/instance

表 5-428 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-429 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	在线迁移任务名称。
description	否	String	在线迁移任务描述。
vpc_id	是	String	虚拟私有云ID。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，在虚拟私有云的详情页面查找VPC ID。 - 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询VPC列表。
subnet_id	是	String	子网的网络ID。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，单击VPC下的子网，进入子网详情页面，查找网络ID。 - 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询子网列表。

参数	是否必选	参数类型	描述
security_group_id	是	String	指定实例所属的安全组。 安全组用来实现安全组内和组间虚拟机的访问控制，加强虚拟机的安全保护。您可以在安全组中定义各种访问规则，当虚拟机加入该安全组后，即受到这些访问规则的保护。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，访问控制安全组选项下可以对安全组进行创建和配置,并获取安全组ID。 - 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询安全组列表。

响应参数

状态码：200

表 5-430 响应 Body 参数

参数	参数类型	描述
instance_id	String	在线迁移任务ID。

状态码：400

表 5-431 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见错误码。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-432 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-433 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-434 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-435 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

创建一个在线数据迁移任务，配置在线迁移虚拟机资源的VPC、子网和安全组。

```
POST https://{dcs_endpoint}/v2/{project_id}/migration/instance
{
  "name" : "dcs-migration-thrc",
  "description" : "创建在线数据迁移任务",
  "vpc_id" : "5e37b3be-950a-48e1-b498-65b63d336481",
  "subnet_id" : "40f15ed6-7f85-49d5-ba0e-65b940d4812c",
  "security_group_id" : "9df96622-24b7-4813-84b8-ab74552a21d7"
}
```

响应示例

状态码：200

创建在线数据迁移任务成功。

```
{
  "instance_id" : "b21989ec-2889-4b8e-99db-19c073425ec2"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个在线数据迁移任务，配置在线迁移虚拟机资源的VPC、子网和安全组。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateOnlineMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateOnlineMigrationTaskRequest request = new CreateOnlineMigrationTaskRequest();
        CreateOnlineMigrationTaskBody body = new CreateOnlineMigrationTaskBody();
        body.withSecurityGroupId("9df96622-24b7-4813-84b8-ab74552a21d7");
        body.withSubnetId("40f15ed6-7f85-49d5-ba0e-65b940d4812c");
        body.withVpcId("5e37b3be-950a-48e1-b498-65b63d336481");
        body.withDescription("创建在线数据迁移任务");
        body.withName("dcs-migration-thrc");
        request.withBody(body);
        try {
            CreateOnlineMigrationTaskResponse response = client.createOnlineMigrationTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建一个在线数据迁移任务，配置在线迁移虚拟机资源的VPC、子网和安全组。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateOnlineMigrationTaskRequest()
    request.body = CreateOnlineMigrationTaskBody(
        security_group_id="9df96622-24b7-4813-84b8-ab74552a21d7",
        subnet_id="40f15ed6-7f85-49d5-ba0e-65b940d4812c",
        vpc_id="5e37b3be-950a-48e1-b498-65b63d336481",
        description="创建在线数据迁移任务",
        name="dcs-migration-thrc"
    )
    response = client.create_online_migration_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

创建一个在线数据迁移任务，配置在线迁移虚拟机资源的VPC、子网和安全组。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.CreateOnlineMigrationTaskRequest{}
descriptionCreateOnlineMigrationTaskBody:= "创建在线数据迁移任务"
request.Body = &model.CreateOnlineMigrationTaskBody{
    SecurityGroupId: "9df96622-24b7-4813-84b8-ab74552a21d7",
    SubnetId: "40f15ed6-7f85-49d5-ba0e-65b940d4812c",
    VpcId: "5e37b3be-950a-48e1-b498-65b63d336481",
    Description: &descriptionCreateOnlineMigrationTaskBody,
    Name: "dcs-migration-thrc",
}
response, err := client.CreateOnlineMigrationTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建在线数据迁移任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.11 配置在线数据迁移任务

功能介绍

配置在线数据迁移任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration/{task_id}/task

表 5-436 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
task_id	是	String	在线迁移任务ID。

请求参数

表 5-437 请求 Body 参数

参数	是否必选	参数类型	描述
migration_method	是	String	迁移方式，包括全量迁移和增量迁移两种类型。 - 全量迁移：该模式为Redis的一次性迁移，适用于可中断业务的迁移场景。 - 增量迁移：该模式为Redis的持续性迁移，适用于对业务中断敏感的迁移场景。 - full_amount_migration：表示全量迁移。 - incremental_migration：表示增量迁移。 全量迁移过程中，如果源Redis有数据更新，这部分更新数据不会被迁移到目标Redis。 增量迁移阶段通过解析日志等技术，持续保持源Redis和目标端Redis的数据一致， 先进行全量迁移，后进行增量迁移。 取值范围：
resume_mode	是	String	自动重连，根据参数决定是否自动重连。 自动重连模式在遇到网络等异常情况时，会无限自动重试。 自动重连模式在无法进行增量同步时，会触发全量同步，增加带宽占用，请谨慎选择。 取值范围： - auto：自动重连。 - manual：手动重连。

参数	是否必选	参数类型	描述
bandwidth_limit_mb	否	String	带宽限制，当迁移方式为增量迁移时，为保证业务正常运行，您可以启用带宽限制功能，当数据同步速度达到带宽限制时，将限制同步速度的继续增长。 - 带宽限制的单位为MB/s。 - 取值范围：1-10240（大于0小于10241之间的整数）。
source_instance	是	ConfigMigrationInstanceBody object	表示源Redis信息。
target_instance	是	ConfigMigrationInstanceBody object	表示目标Redis信息。

表 5-438 ConfigMigrationInstanceBody

参数	是否必选	参数类型	描述
id	否	String	Redis实例ID（若不填写addr，则必须填写ID）。
addr	否	String	Redis实例地址（若不填写ID，则必须填写addr）。
password	否	String	Redis密码，如果设置了密码，则必须填写。

响应参数

状态码：200

表 5-439 响应 Body 参数

参数	参数类型	描述
id	String	迁移任务ID。
name	String	迁移任务名称。
status	String	迁移任务状态，这个字段的值包括：SUCCESS（成功），FAILED（失败），MIGRATING（迁移中），TERMINATED（已结束）。
error	String	错误信息。

状态码：400

表 5-440 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-441 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-442 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-443 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-444 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

配置在线迁移任务，迁移方式为增量迁移，开启自动重连，配置源端和目标端实例信息。

```
POST https://{dcs_endpoint}/v2/{project_id}/migration/{task_id}/task
{
  "migration_method": "incremental_migration",
  "bandwidth_limit_mb": 123,
  "resume_mode": "auto",
  "source_instance": {
    "id": null,
    "addrs": "192.168.1.1:6379,192.168.0.0:6379",
    "password": "xxxxxx"
  },
  "target_instance": {
    "id": "cf4a05df-1c38-47c5-bb5a-0a7b3673b3bd",
    "addrs": null,
    "password": null
  }
}
```

响应示例

状态码：200

配置在线数据迁移任务成功。

```
{
  "id": "90754308-a156-406f-a837-8f852f38a646",
  "name": "dcs-migration-1db7",
  "status": "FULLMIGRATING",
  "error": ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

配置在线迁移任务，迁移方式为增量迁移，开启自动重连，配置源端和目标端实例信息。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class SetOnlineMigrationTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        SetOnlineMigrationTaskRequest request = new SetOnlineMigrationTaskRequest();
        request.withTaskId("{task_id}");
        SetOnlineMigrationTaskBody body = new SetOnlineMigrationTaskBody();
        ConfigMigrationInstanceBody targetInstancebody = new ConfigMigrationInstanceBody();
        targetInstancebody.withId("cf4a05df-1c38-47c5-bb5a-0a7b3673b3bd");
        ConfigMigrationInstanceBody sourceInstancebody = new ConfigMigrationInstanceBody();
        sourceInstancebody.withAddr("192.168.1.1:6379,192.168.0.0:6379")
            .withPassword("xxxxxx");
        body.withTargetInstance(targetInstancebody);
        body.withSourceInstance(sourceInstancebody);
        body.withBandwidthLimitMb("123");
        body.withResumeMode(SetOnlineMigrationTaskBody.ResumeModeEnum.fromValue("auto"));

        body.withMigrationMethod(SetOnlineMigrationTaskBody.MigrationMethodEnum.fromValue("incremental_migration"));
        request.withBody(body);
        try {
            SetOnlineMigrationTaskResponse response = client.setOnlineMigrationTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

配置在线迁移任务，迁移方式为增量迁移，开启自动重连，配置源端和目标端实例信息。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SetOnlineMigrationTaskRequest()
        request.task_id = "{task_id}"
        targetInstancebody = ConfigMigrationInstanceBody(
            id="cf4a05df-1c38-47c5-bb5a-0a7b3673b3bd"
        )
        sourceInstancebody = ConfigMigrationInstanceBody(
            addrs="192.168.1.1:6379,192.168.0.0:6379",
            password="xxxxxx"
        )
        request.body = SetOnlineMigrationTaskBody(
            target_instance=targetInstancebody,
            source_instance=sourceInstancebody,
            bandwidth_limit_mb="123",
            resume_mode="auto",
            migration_method="incremental_migration"
        )
        response = client.set_online_migration_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

配置在线迁移任务，迁移方式为增量迁移，开启自动重连，配置源端和目标端实例信息。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SetOnlineMigrationTaskRequest{}
    request.TaskId = "{task_id}"
    idTargetInstance := "cf4a05df-1c38-47c5-bb5a-0a7b3673b3bd"
    targetInstancebody := &model.ConfigMigrationInstanceBody{
        Id: &idTargetInstance,
    }
    addrSourceInstance := "192.168.1.1:6379,192.168.0.0:6379"
    passwordSourceInstance := "xxxxxx"
    sourceInstancebody := &model.ConfigMigrationInstanceBody{
        Addrs: &addrSourceInstance,
        Password: &passwordSourceInstance,
    }
    bandwidthLimitMbSetOnlineMigrationTaskBody := "123"
    request.Body = &model.SetOnlineMigrationTaskBody{
        TargetInstance: targetInstancebody,
        SourceInstance: sourceInstancebody,
        BandwidthLimitMb: &bandwidthLimitMbSetOnlineMigrationTaskBody,
        ResumeMode: model.GetSetOnlineMigrationTaskBodyResumeModeEnum().AUTO,
        MigrationMethod:
    model.GetSetOnlineMigrationTaskBodyMigrationMethodEnum().INCREMENTAL_MIGRATION,
    }
    response, err := client.SetOnlineMigrationTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	配置在线数据迁移任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.12 批量停止数据迁移任务

功能介绍

批量停止数据迁移任务，接口响应成功，仅表示下发任务成功。查询到迁移任务状态为TERMINATED时，即停止成功。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task/batch-stop

表 5-445 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-446 请求 Body 参数

参数	是否必选	参数类型	描述
migration_tasks	是	Array of strings	数据迁移任务列表。

响应参数

状态码：200

表 5-447 响应 Body 参数

参数	参数类型	描述
migration_tasks	Array of StopMigrationTaskResult objects	数据迁移任务列表。

表 5-448 StopMigrationTaskResult

参数	参数类型	描述
result	String	下发停止迁移任务操作结果。
task_id	String	数据迁移任务ID。

状态码：400

表 5-449 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-450 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-451 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-452 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-453 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

批量停止数据迁移任务。

```
POST https://{dcs_endpoint}/v2/{project_id}/migration-task/batch-stop

{
  "migration_tasks": [ "b21989ec-2889-4b8e-99db-19c073425ec2", "5130d57f-640a-435b-
bc3a-0fb1860a5340" ]
}
```

响应示例

状态码：200

批量下发停止迁移任务成功。

说明

查询迁移任务状态为TERMINATED，即停止成功。

```
{
  "migration_tasks": [ {
    "result": "success",
    "task_id": "b21989ec-2889-4b8e-99db-19c073425ec2"
  }, {
    "result": "failed",
    "task_id": "5130d57f-640a-435b-bc3a-0fb1860a5340"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_msg": "invalid migration task id in the request.",
  "error_code": "DCS.4855"
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量停止数据迁移任务。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchStopMigrationTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    }
}
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
BatchStopMigrationTasksRequest request = new BatchStopMigrationTasksRequest();
BatchStopMigrationTasksBody body = new BatchStopMigrationTasksBody();
List<String> listbodyMigrationTasks = new ArrayList<>();
listbodyMigrationTasks.add("b21989ec-2889-4b8e-99db-19c073425ec2");
listbodyMigrationTasks.add("5130d57f-640a-435b-bc3a-0fb1860a5340");
body.withMigrationTasks(listbodyMigrationTasks);
request.withBody(body);
try {
    BatchStopMigrationTasksResponse response = client.batchStopMigrationTasks(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

批量停止数据迁移任务。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchStopMigrationTasksRequest()
        listMigrationTasksbody = [
```

```
        "b21989ec-2889-4b8e-99db-19c073425ec2",  
        "5130d57f-640a-435b-bc3a-0fb1860a5340"  
    ]  
    request.body = BatchStopMigrationTasksBody(  
        migration_tasks=listMigrationTasksbody  
    )  
    response = client.batch_stop_migration_tasks(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

批量停止数据迁移任务。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dcs.NewDcsClient(  
        dcs.DcsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.BatchStopMigrationTasksRequest{  
        var listMigrationTasksbody = []string{  
            "b21989ec-2889-4b8e-99db-19c073425ec2",  
            "5130d57f-640a-435b-bc3a-0fb1860a5340",  
        }  
        request.Body = &model.BatchStopMigrationTasksBody{  
            MigrationTasks: listMigrationTasksbody,  
        }  
        response, err := client.BatchStopMigrationTasks(request)  
        if err == nil {  
            fmt.Printf("%+v\n", response)  
        } else {  
            fmt.Println(err)  
        }  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	批量下发停止迁移任务成功。 说明 查询迁移任务状态为TERMINATED，即停止成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.13 同步停止数据迁移任务

功能介绍

同步停止数据迁移任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-task/{task_id}/sync-stop

表 5-454 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
task_id	是	String	任务ID。

请求参数

无

响应参数

状态码：204
同步停止数据迁移任务成功。

状态码：400

表 5-455 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-456 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-457 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-458 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-459 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class StopMigrationTaskSyncSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        StopMigrationTaskSyncRequest request = new StopMigrationTaskSyncRequest();
        request.withTaskId("{task_id}");
        try {
            StopMigrationTaskSyncResponse response = client.stopMigrationTaskSync(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = StopMigrationTaskSyncRequest()
    request.task_id = "{task_id}"
    response = client.stop_migration_task_sync(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopMigrationTaskSyncRequest{
        request.TaskId = "{task_id}"
    }
    response, err := client.StopMigrationTaskSync(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	同步停止数据迁移任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.14 查询迁移日志列表

功能介绍

查询迁移日志列表

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/migration-task/{task_id}/logs

表 5-460 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 数据迁移任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-461 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 10
log_level	否	String	参数解释： 日志级别。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-462 响应 Body 参数

参数	参数类型	描述
log_num	Integer	参数解释： 日志条数。 取值范围： 不涉及。
migration_logs	Array of MigrationLog objects	参数解释： 日志列表。

表 5-463 MigrationLog

参数	参数类型	描述
created_at	String	参数解释： 迁移日志生成时间，形如： 2023-05-15T09:11:25.449Z。 取值范围： 不涉及。
log_level	String	参数解释： 日志级别。 取值范围： • INFO: 提示日志。 • ERROR: 错误日志。
message	String	参数解释： 日志信息。 取值范围： 不涉及。
log_code	String	参数解释： 日志的编码。 取值范围： 不涉及。
keyword	Array of strings	参数解释： 日志中的关键字。 取值范围： 不涉及。

状态码：400

表 5-464 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-465 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-466 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-467 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-468 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{tenant_id}/migration-task/{task_id}/logs?offset=1&limit=10&log_level=info
```

响应示例

状态码：200

查询迁移日志列表成功。

```
{
  "log_num" : 2,
  "migration_logs" : [ {
    "created_at" : "2023-05-15T09:14:40.000Z",
    "keyword" : [ "8abfa7aa87ef158901881ead51a47dcd" ],
    "log_code" : "migration.0012",
    "log_level" : "INFO",
    "message" : "job success, all files import complete.Job id: 8abfa7aa87ef158901881ead51a47dcd."
  }, {
    "created_at" : "2023-05-15T09:14:38.000Z",
    "keyword" : [ "DCS_Cache_Hidden_Image_3.0.10.tar.gz", "1/1" ],
    "log_code" : "migration.0026",
    "log_level" : "INFO",
    "message" : "Migrating files succeeded. File name: DCS_Cache_Hidden_Image_3.0.10.tar.gz; progress: 1/1."
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListMigrationTaskLogsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListMigrationTaskLogsRequest request = new ListMigrationTaskLogsRequest();
        request.withTaskId("{task_id}");
        try {
            ListMigrationTaskLogsResponse response = client.listMigrationTaskLogs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListMigrationTaskLogsRequest()
        request.task_id = "{task_id}"
        response = client.list_migration_task_logs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListMigrationTaskLogsRequest{}
    request.TaskId = "{task_id}"
    response, err := client.ListMigrationTaskLogs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询迁移日志列表成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.6.15 批量重启在线迁移任务

功能介绍

批量重启在线迁移任务，接口响应成功，返回重启在线迁移任务下发结果。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/migration-tasks/batch-restart

表 5-469 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-470 请求 Body 参数

参数	是否必选	参数类型	描述
migration_tasks	是	Array of strings	数据迁移任务列表。

响应参数

状态码：200

表 5-471 响应 Body 参数

参数	参数类型	描述
migration_tasks	Array of BatchRestartMigrationTaskResult objects	数据迁移任务列表。

表 5-472 BatchRestartMigrationTaskResult

参数	参数类型	描述
result	String	下发重启迁移任务操作结果。
task_id	String	数据迁移任务ID。
error_msg	String	错误信息
error_code	String	错误码

状态码：400

表 5-473 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-474 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-475 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-476 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-477 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/migration-tasks/batch-restart
{
  "migration_tasks": [ "091434c0-d6d7-47c6-9e14-17cc9502b27b", "7599c3ed-
d0d3-4fb9-9273-653160302023" ]
}
```

响应示例

状态码：200
批量重启在线迁移任务成功

```
{
  "migration_tasks": [ {
    "result": "success",
    "task_id": "7599c3ed-d0d3-4fb9-9273-653160302023"
  }, {
    "result": "failed",
    "task_id": "091434c0-d6d7-47c6-9e14-17cc9502b27b",
    "error_code": "dcs.41.0001",
    "error_msg": "task not v2"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```



```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchRestartOnlineMigrationTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        BatchRestartOnlineMigrationTasksRequest request = new BatchRestartOnlineMigrationTasksRequest();
        BatchRestartOnlineMigrationTasksBody body = new BatchRestartOnlineMigrationTasksBody();
        List<String> listbodyMigrationTasks = new ArrayList<>();
        listbodyMigrationTasks.add("091434c0-d6d7-47c6-9e14-17cc9502b27b");
        listbodyMigrationTasks.add("7599c3ed-d0d3-4fb9-9273-653160302023");
        body.withMigrationTasks(listbodyMigrationTasks);
        request.withBody(body);
        try {
            BatchRestartOnlineMigrationTasksResponse response =
client.batchRestartOnlineMigrationTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchRestartOnlineMigrationTasksRequest()
    listMigrationTasksbody = [
        "091434c0-d6d7-47c6-9e14-17cc9502b27b",
        "7599c3ed-d0d3-4fb9-9273-653160302023"
    ]
    request.body = BatchRestartOnlineMigrationTasksBody(
        migration_tasks=listMigrationTasksbody
    )
    response = client.batch_restart_online_migration_tasks(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRestartOnlineMigrationTasksRequest{}
    var listMigrationTasksbody = []string{
        "091434c0-d6d7-47c6-9e14-17cc9502b27b",
```

```
    "7599c3ed-d0d3-4fb9-9273-653160302023",
  }
  request.Body = &model.BatchRestartOnlineMigrationTasksBody{
    MigrationTasks: listMigrationTasksbody,
  }
  response, err := client.BatchRestartOnlineMigrationTasks(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	批量重启在线迁移任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.7 标签管理

5.7.1 查询租户所有标签

功能介绍

查询租户在指定Project中实例类型的所有资源标签集合。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/dcs/tags

表 5-478 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-479 响应 Body 参数

参数	参数类型	描述
tags	Array of Tag objects	参数解释： 标签列表。 取值范围： 不涉及。

表 5-480 Tag

参数	参数类型	描述
key	String	参数解释： 标签键，最大长度128个字符。 取值范围： 不涉及。
values	Array of strings	参数解释： 标签值。 取值范围： 不涉及。

状态码：400

表 5-481 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-482 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

POST https://{dcs_endpoint}/v2/{project_id}/dcs/tags

响应示例

状态码：200

查询租户所有标签成功。

```
{
  "tags": [ {
    "values": [ "value1", "value2" ],
    "key": "1"
  }, {
    "values": [ "value1", "value2" ],
    "key": "2"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListTagsOfTenantSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTagsOfTenantRequest request = new ListTagsOfTenantRequest();
        try {
            ListTagsOfTenantResponse response = client.listTagsOfTenant(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTagsOfTenantRequest()
        response = client.list_tags_of_tenant(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListTagsOfTenantRequest{}
response, err := client.ListTagsOfTenant(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询租户所有标签成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.7.2 批量添加或删除标签

功能介绍

为指定实例批量添加标签，或批量删除标签。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/dcs/{instance_id}/tags/action

表 5-483 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-484 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	参数解释： 操作标识。 约束限制： 不涉及。 取值范围： - create：创建实例标签。 - delete：删除实例标签。 默认取值： 不涉及。
tags	是	Array of ResourceTag objects	参数解释： 标签列表。 约束限制： 不涉及。

表 5-485 ResourceTag

参数	是否必选	参数类型	描述
key	是	String	参数解释： 标签键。 约束限制： • 对于同一资源键值唯一。 • 长度不超过128个字符。 • 标签的键必须唯一且输入不能为空。 取值范围： 不涉及。 默认取值： 不涉及。
value	否	String	参数解释： 标签值。 约束限制： • action为create时必选。 action为delete时非必选。 • 长度不超过255个字符。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：204

参数解释：

添加/删除标签成功。

状态码：400

表 5-486 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-487 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-488 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-489 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-490 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- 为指定实例批量添加标签。

```
POST https://{dcs_endpoint}/v2/7dddae81f0e34f62adb9618bc8c8b1fe/dcs/01928d55-7020-4500-9c29-774caabe4bc4/tags/action
{
  "action": "create",
  "tags": [ {
    "value": "2",
    "key": "dcs001"
  }, {
    "value": "4",
    "key": "dcs003"
  } ]
}
```
- 为指定实例批量删除标签。

```
POST https://{dcs_endpoint}/v2/7dddae81f0e34f62adb9618bc8c8b1fe/dcs/01928d55-7020-4500-9c29-774caabe4bc4/tags/action
{
  "action": "delete",
  "tags": [ {
    "key": "key1",
    "value": "11"
  } ]
}
```

响应示例

状态码：400

参数解释：

非法请求。

```
{
  "error_code" : "DCS.4005",
  "error_msg" : "The message body contains invalid characters or is not in JSON format."
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 为指定实例批量添加标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateOrDeleteTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateOrDeleteTagsRequest request = new BatchCreateOrDeleteTagsRequest();
        request.withInstanceId("{instance_id}");
        CreateOrDeleteInstanceTags body = new CreateOrDeleteInstanceTags();
        List<ResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTag()
                .withKey("dcs001")
                .withValue("2")
        );
        listbodyTags.add(
            new ResourceTag()
                .withKey("dcs003")
                .withValue("4")
        );
        body.withTags(listbodyTags);
        body.withAction("create");
        request.withBody(body);
        try {
            BatchCreateOrDeleteTagsResponse response = client.batchCreateOrDeleteTags(request);
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

- 为指定实例批量删除标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateOrDeleteTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateOrDeleteTagsRequest request = new BatchCreateOrDeleteTagsRequest();
        request.withInstanceId("{instance_id}");
        CreateOrDeleteInstanceTags body = new CreateOrDeleteInstanceTags();
        List<ResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new ResourceTag()
                .withKey("key1")
                .withValue("11")
        );
        body.withTags(listbodyTags);
        body.withAction("delete");
        request.withBody(body);
        try {
            BatchCreateOrDeleteTagsResponse response = client.batchCreateOrDeleteTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- 为指定实例批量添加标签。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateOrDeleteTagsRequest()
        request.instance_id = "{instance_id}"
        listTagsbody = [
            ResourceTag(
                key="dcs001",
                value="2"
            ),
            ResourceTag(
                key="dcs003",
                value="4"
            )
        ]
        request.body = CreateOrDeleteInstanceTags(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_or_delete_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 为指定实例批量删除标签。

```
# coding: utf-8
```



```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateOrDeleteTagsRequest()
        request.instance_id = "{instance_id}"
        listTagsbody = [
            ResourceTag(
                key="key1",
                value="11"
            )
        ]
        request.body = CreateOrDeleteInstanceTags(
            tags=listTagsbody,
            action="delete"
        )
        response = client.batch_create_or_delete_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 为指定实例批量添加标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchCreateOrDeleteTagsRequest{
    request.InstanceId = "{instance_id}"
    valueTags:= "2"
    valueTags1:= "4"
    var listTagsbody = []model.ResourceTag{
        {
            Key: "dcs001",
            Value: &valueTags,
        },
        {
            Key: "dcs003",
            Value: &valueTags1,
        },
    }
    request.Body = &model.CreateOrDeleteInstanceTags{
        Tags: listTagsbody,
        Action: "create",
    }
    response, err := client.BatchCreateOrDeleteTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 为指定实例批量删除标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
Build()

request := &model.BatchCreateOrDeleteTagsRequest{}
request.InstanceId = "{instance_id}"
valueTags:= "11"
var listTagsbody = []model.ResourceTag{
    {
        Key: "key1",
        Value: &valueTags,
    },
}
request.Body = &model.CreateOrDeleteInstanceTags{
    Tags: listTagsbody,
    Action: "delete",
}
response, err := client.BatchCreateOrDeleteTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	参数解释： 添加/删除标签成功。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.7.3 查询单个实例标签

功能介绍

通过实例ID查询标签。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/tags

表 5-491 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-492 响应 Body 参数

参数	参数类型	描述
tags	Array of QueryResourceTag objects	参数解释： 标签列表。

表 5-493 QueryResourceTag

参数	参数类型	描述
key	String	参数解释： 标签键。 取值范围： 不涉及。
value	String	参数解释： 标签值。 取值范围： 不涉及。

状态码：400

表 5-494 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-495 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-496 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-497 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-498 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/tags
```

响应示例

状态码：200

参数解释：

查询实例标签成功。

```
{
  "tags": [ {
    "value": "a",
    "key": "1"
  }, {
    "value": "b",
    "key": "2"
  } ]
}
```

状态码：400

参数解释：

非法请求。

```
{
  "error_code": "DCS.4005",
  "error_msg": "The message body contains invalid characters or is not in JSON format."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
```



```
        .build();
        ShowTagsRequest request = new ShowTagsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowTagsResponse response = client.showTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTagsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTagsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 查询实例标签成功。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.8 缓存分析

5.8.1 查询过期 Key 扫描记录

功能介绍

查询过期Key扫描记录。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/auto-expire/histories

表 5-499 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-500 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10
status	否	String	参数解释： 过期key扫描任务的状态。 约束限制： 不涉及。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。 默认取值： 不涉及。

请求参数

无

响应参数

状态码： 200

表 5-501 响应 Body 参数

参数	参数类型	描述
records	Array of SimpleKeyScanRecord objects	参数解释： 过期key记录。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
total	Integer	参数解释： 扫描过期key的总数。 取值范围： 不涉及。
count	Integer	参数解释： 扫描过期key的次数。 取值范围： 不涉及。

表 5-502 SimpleKeyScanRecord

参数	参数类型	描述
id	String	参数解释： 过期key扫描任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 过期key扫描任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。

参数	参数类型	描述
scan_type	String	参数解释： 扫描方式。 取值范围： • auto：自动扫描。 • manual：手动扫描。
num	Integer	参数解释： 一次扫描过期key的数量。 取值范围： 不涉及。
created_at	String	参数解释： 扫描任务创建时间，格式为： 2023-06-13T13:46:14.771Z。 取值范围： 不涉及。
started_at	String	参数解释： 扫描任务开始时间，格式为： 2023-06-13T13:46:14.771Z。 取值范围： 不涉及。
finished_at	String	参数解释： 扫描任务完成时间，格式为： 2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-503 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-504 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/auto-expire/histories
```

响应示例

状态码：200

查询过期Key扫描记录成功。

```
{  
  "records" : [ {
```

```
"id" : "8476c93a-c3bc-494a-924c-29a2e71ea0d9",
"instance_id" : "69f99808-34cd-4ce6-ae73-fd03b261b5f3",
"status" : "running",
"scan_type" : "auto",
"num" : 100,
"created_at" : "2023-06-13T13:46:14.771Z",
"started_at" : "2023-06-13T13:46:15.113Z",
"finished_at" : ""
}],
"instance_id" : "69f99808-34cd-4ce6-ae73-fd03b261b5f3",
"total" : 1,
"count" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowExpireKeyScanInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowExpireKeyScanInfoRequest request = new ShowExpireKeyScanInfoRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowExpireKeyScanInfoResponse response = client.showExpireKeyScanInfo(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```



```
}
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowExpireKeyScanInfoRequest()
        request.instance_id = "{instance_id}"
        response = client.show_expire_key_scan_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```
client := dcs.NewDcsClient(  
    dcs.DcsClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ShowExpireKeyScanInfoRequest{  
    request.InstanceId = "{instance_id}"  
}  
response, err := client.ShowExpireKeyScanInfo(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询过期Key扫描记录成功。
400	非法请求。
500	内部服务器错误。

错误码

请参见[错误码](#)。

5.8.2 立刻扫描过期 Key

功能介绍

立刻扫描过期Key。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/auto-expire/scan

表 5-505 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-506 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 过期key扫描任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 过期key扫描任务的状态。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。
scan_type	String	参数解释： 扫描方式。 取值范围： • auto：自动扫描。 • manual：手动扫描。
created_at	String	参数解释： 扫描任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 扫描任务开始时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
finished_at	String	参数解释： 扫描任务完成时间, 格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。

状态码：400

表 5-507 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-508 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/auto-expire/scan
```

响应示例

状态码：200

立刻扫描过期Key成功。

```
{  
  "id": "8f3620cc-65b0-43d4-9bd6-4b60a35b5559",  
}
```

```
"instance_id" : "56776da5-761a-4c7a-8acf-6646b23a2978",
"status" : "waiting",
"scan_type" : "manual",
"created_at" : "2023-06-13T13:49:35.058Z"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ScanExpireKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ScanExpireKeyRequest request = new ScanExpireKeyRequest();
        request.withInstanceId("{instance_id}");
        try {
            ScanExpireKeyResponse response = client.scanExpireKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ScanExpireKeyRequest()
        request.instance_id = "{instance_id}"
        response = client.scan_expire_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ScanExpireKeyRequest{}
```

```
request.InstanceId = "{instance_id}"
response, err := client.ScanExpireKey(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	立刻扫描过期Key成功。
400	非法请求。
500	内部服务器错误。

错误码

请参见[错误码](#)。

5.8.3 创建大 key 分析任务

功能介绍

为Redis实例创建大key分析任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/bigkey-task

表 5-509 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-510 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 大key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 大key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。
scan_type	String	参数解释： 分析方式。 取值范围： - manual：手动分析。 - auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为："2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为："2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为："2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
num	Integer	参数解释： 大key的数量。 取值范围： 不涉及。
keys	Array of BigkeysBody objects	参数解释： 大key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-511 BigkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： • string：字符串类型。 • list：列表类型。 • set：集合类型。 • zset：有序类型。 • hash：哈希类型。
shard	String	参数解释： 大key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 大key所在的DB。 取值范围： 0-255
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： • count：key的数量。 • byte：key的大小。

状态码：400

表 5-512 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-513 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-514 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-515 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-516 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-  
cebf-4473-95c4-d1b573c16e79/bigkey-task
```

响应示例

状态码：200

创建大key分析任务成功。

```
{  
  "id" : "858ee14c-2271-4489-8b82-7bda7459ae3e",  
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",  
  "status" : "waiting",  
  "scan_type" : "manual",  
  "created_at" : "2020-06-15T02:21:18.669Z",  
  "num" : 0  
}
```

状态码：400

非法请求。

```
{  
  "error_code" : "DCS.4919",  
  "error_msg" : "Does not support bigkey analyze."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateBigkeyScanTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateBigkeyScanTaskRequest request = new CreateBigkeyScanTaskRequest();
        request.withInstanceId("{instance_id}");
        try {
            CreateBigkeyScanTaskResponse response = client.createBigkeyScanTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateBigkeyScanTaskRequest()
    request.instance_id = "{instance_id}"
    response = client.create_bigkey_scan_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateBigkeyScanTaskRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.CreateBigkeyScanTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建大key分析任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.4 查询大 key 分析任务列表

功能介绍

查询大key分析任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/bigkey-tasks

表 5-517 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-518 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10

参数	是否必选	参数类型	描述
status	否	String	参数解释： 过期key扫描任务的状态。 约束限制： 不涉及。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-519 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
count	Integer	参数解释： 大key分析的次数。 取值范围： 不涉及。
records	Array of RecordsResponse objects	参数解释： 大key分析记录列表。 取值范围： 不涉及。

表 5-520 RecordsResponse

参数	参数类型	描述
id	String	参数解释： 分析操作的任务ID。 取值范围： 不涉及。
status	String	参数解释： 热key分析任务的状态。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建的时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务的开始时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务的结束时间，式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。

状态码：400

表 5-521 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-522 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-523 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-524 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-525 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/bigkey-tasks
```

响应示例

状态码：200

查询大key分析历史记录。

```
{
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "count" : 1,
  "records" : [ {
    "id" : "858ee14c-2271-4489-8b82-7bda7459ae3e",
    "scan_type" : "manual",
    "status" : "success",
    "created_at" : "2020-06-15T02:21:18.669Z",
    "started_at" : "2020-06-15T02:21:23.534Z",
    "finished_at" : "2020-06-15T02:21:25.588Z"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4919",
  "error_msg" : "Does not support bigkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListBigkeyScanTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBigkeyScanTasksRequest request = new ListBigkeyScanTasksRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListBigkeyScanTasksResponse response = client.listBigkeyScanTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListBigkeyScanTasksRequest()
    request.instance_id = "{instance_id}"
    response = client.list_bigkey_scan_tasks(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBigkeyScanTasksRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListBigkeyScanTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询大key分析历史记录。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.5 查询大 key 分析详情

功能介绍

查询大key分析详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/bigkey-task/{bigkey_id}

表 5-526 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
bigkey_id	是	String	参数解释： 大key分析任务ID。可调用 查询大key分析任务列表 接口，通过响应参数id的返回值获取。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-527 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 大key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 大key分析任务的状态。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建分 析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建分 析任务时此值为null，不返回）。 取值范围： 不涉及。

参数	参数类型	描述
num	Integer	参数解释： 大key的数量。 取值范围： 不涉及。
keys	Array of BigkeysBody objects	参数解释： 大key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-528 BigkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： • string：字符串类型。 • list：列表类型。 • set：集合类型。 • zset：有序类型。 • hash：哈希类型。
shard	String	参数解释： 大key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 大key所在的DB。 取值范围： 0-255

参数	参数类型	描述
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： - count: key的数量。 - byte: key的大小。

状态码：400

表 5-529 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-530 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-531 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-532 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-533 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-  
cebf-4473-95c4-d1b573c16e79/bigkey-task/858ee14c-2271-4489-8b82-7bda7459ae3e
```


响应示例

状态码：200

查询大key分析详情成功。

```
{
  "id": "858ee14c-2271-4489-8b82-7bda7459ae3e",
  "instance_id": "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "status": "success",
  "scan_type": "manual",
  "created_at": "2020-06-15T02:21:18.669Z",
  "started_at": "2020-06-15T02:21:23.534Z",
  "finished_at": "2020-06-15T02:21:25.588Z",
  "keys": [ {
    "name": "dcs-bigkey-test",
    "type": "string",
    "shard": "192.168.1.134:6379",
    "db": 0,
    "size": 3000,
    "unit": "byte"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4919",
  "error_msg": "Does not support bigkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowBigkeyScanTaskDetailsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowBigkeyScanTaskDetailsRequest request = new ShowBigkeyScanTaskDetailsRequest();
request.withInstanceId("{instance_id}");
request.withBigkeyId("{bigkey_id}");
try {
    ShowBigkeyScanTaskDetailsResponse response = client.showBigkeyScanTaskDetails(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBigkeyScanTaskDetailsRequest()
        request.instance_id = "{instance_id}"
        request.bigkey_id = "{bigkey_id}"
        response = client.show_bigkey_scan_task_details(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBigkeyScanTaskDetailsRequest{}
    request.InstanceId = "{instance_id}"
    request.BigkeyId = "{bigkey_id}"
    response, err := client.ShowBigkeyScanTaskDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询大key分析详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.6 删除大 key 分析记录

功能介绍

删除大key分析记录。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/bigkey-task/{bigkey_id}

表 5-534 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
bigkey_id	是	String	参数解释： 大key分析任务ID。可调用 查询大key分析任务列表 接口，通过响应参数id的返回值获取。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-535 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 大key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 大key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。

参数	参数类型	描述
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
num	Integer	参数解释： 大key的数量。 取值范围： 不涉及。
keys	Array of BigkeysBody objects	参数解释： 大key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-536 BigkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： • string：字符串类型。 • list：列表类型。 • set：集合类型。 • zset：有序类型。 • hash：哈希类型。
shard	String	参数解释： 大key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 大key所在的DB。 取值范围： 0-255
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： • count：key的数量。 • byte：key的大小。

状态码：204
无大key分析任务被删除。

状态码：400

表 5-537 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-538 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-539 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-540 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-541 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/bigkey-task/858ee14c-2271-4489-8b82-7bda7459ae3e
```

响应示例

状态码：200

删除大key分析任务成功。

```
{
  "id": "858ee14c-2271-4489-8b82-7bda7459ae3e",
  "instance_id": "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "status": "success",
  "scan_type": "manual",
  "created_at": "2020-06-15T02:21:18.669Z",
  "started_at": "2020-06-15T02:21:23.534Z",
  "finished_at": "2020-06-15T02:21:25.588Z",
  "keys": [ {
    "name": "dcs-bigkey-test",
    "type": "string",
    "shard": "192.168.1.134:6379",
    "db": 0,
    "size": 3000,
    "unit": "byte"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4919",
  "error_msg": "Does not support bigkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteBigkeyScanTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBigkeyScanTaskRequest request = new DeleteBigkeyScanTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withBigkeyId("{bigkey_id}");
        try {
            DeleteBigkeyScanTaskResponse response = client.deleteBigkeyScanTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBigkeyScanTaskRequest()
        request.instance_id = "{instance_id}"
        request.bigkey_id = "{bigkey_id}"
        response = client.delete_bigkey_scan_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBigkeyScanTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.BigkeyId = "{bigkey_id}"
    response, err := client.DeleteBigkeyScanTask(request)
    if err == nil {
```

```
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除大key分析任务成功。
204	无大key分析任务被删除。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.7 设置大 key 自动分析配置

功能介绍

设置大key自动分析配置。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/bigkey/autoscan

表 5-542 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-543 请求 Body 参数

参数	是否必选	参数类型	描述
enable_auto_scan	是	Boolean	参数解释： 是否开启缓存分析定时任务。 约束限制： 不涉及。 取值范围： • true • false 默认取值： 不涉及。

参数	是否必选	参数类型	描述
schedule_at	是	Array of strings	参数解释： 每日缓存分析时间，时间格式为21:00，时间为UTC时间。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-544 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启缓存分析定时任务。 取值范围： - true - false
schedule_at	Array of strings	参数解释： 每日分析时间，时间格式为21:00，时间为UTC时间。 取值范围： 不涉及。
updated_at	String	参数解释： 配置更新时间，时间格式为2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-545 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-546 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-547 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-548 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-549 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

设置大key自动分析任务，每日缓存分析时间为21：00。

```
PUT https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-
cebf-4473-95c4-d1b573c16e79/bigkey/autoscan
{
  "enable_auto_scan": true,
  "schedule_at": [ "21:00" ]
}
```

响应示例

状态码：200

设置大key分析配置成功。

```
{
  "instance_id": "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "enable_auto_scan": true,
  "schedule_at": [ "21:00" ],
  "updated_at": "2020-06-17T02:42:40.793Z"
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4919",
  "error_msg": "Does not support bigkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置大key自动分析任务，每日缓存分析时间为21：00。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateBigkeyAutoscanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateBigkeyAutoscanConfigRequest request = new UpdateBigkeyAutoscanConfigRequest();
        request.withInstanceId("{instance_id}");
        AutoscanConfigRequest body = new AutoscanConfigRequest();
        List<String> listbodyScheduleAt = new ArrayList<>();
        listbodyScheduleAt.add("21:00");
        body.withScheduleAt(listbodyScheduleAt);
        body.withEnableAutoScan(true);
        request.withBody(body);
        try {
            UpdateBigkeyAutoscanConfigResponse response = client.updateBigkeyAutoscanConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

设置大key自动分析任务，每日缓存分析时间为21：00。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateBigkeyAutoscanConfigRequest()
        request.instance_id = "{instance_id}"
        listScheduleAtbody = [
            "21:00"
        ]
        request.body = AutoscanConfigRequest(
            schedule_at=listScheduleAtbody,
            enable_auto_scan=True
        )
        response = client.update_bigkey_autoscan_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

设置大key自动分析任务，每日缓存分析时间为21：00。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateBigkeyAutoscanConfigRequest{}
request.InstanceId = "{instance_id}"
var listScheduleAtbody = []string{
    "21:00",
}
request.Body = &model.AutoscanConfigRequest{
    ScheduleAt: listScheduleAtbody,
    EnableAutoScan: true,
}
response, err := client.UpdateBigkeyAutoscanConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	设置大key分析配置成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.8 查询大 key 自动分析配置

功能介绍

查询大key自动分析配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/bigkey/autoscan

表 5-550 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-551 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启缓存分析定时任务。 取值范围： • true • false
schedule_at	Array of strings	参数解释： 每日分析时间，时间格式为21:00，时间为UTC时间。 取值范围： 不涉及。
updated_at	String	参数解释： 配置更新时间，时间格式为2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-552 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-553 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-554 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-555 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-556 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/bigkey/autoscan
```

响应示例

状态码：200

查询大key自动分析配置。

```
{
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "enable_auto_scan" : true,
  "schedule_at" : [ "21:00" ],
  "updated_at" : "2020-06-17T02:42:40.793Z"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4919",
  "error_msg" : "Does not support bigkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowBigkeyAutoscanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBigkeyAutoscanConfigRequest request = new ShowBigkeyAutoscanConfigRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowBigkeyAutoscanConfigResponse response = client.showBigkeyAutoscanConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowBigkeyAutoscanConfigRequest()
    request.instance_id = "{instance_id}"
    response = client.show_bigkey_autoscan_config(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBigkeyAutoscanConfigRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowBigkeyAutoscanConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询大key自动分析配置。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.9 创建热 key 分析任务

功能介绍

创建热key分析任务，Redis 3.0不支持热key分析。

热key分析需要将缓存实例配置参数maxmemory-policy设置为allkeys-lfu或volatile-lfu。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/hotkey-task

表 5-557 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-558 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 热key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 热key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。

参数	参数类型	描述
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
num	Integer	参数解释： 热key的数量。 取值范围： 不涉及。
keys	Array of HotkeysBody objects	参数解释： 热key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-559 HotkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： • string：字符串类型。 • list：列表类型。 • set：集合类型。 • zset：有序类型。 • hash：哈希类型。
shard	String	参数解释： 热key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 热key所在的DB。 取值范围： 0-255
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： • count：key的数量。 • byte：key的大小。

参数	参数类型	描述
freq	Integer	参数解释： 表示某个key在一段时间的访问频度，会随着访问的频率而变化。 该值并不是简单的访问频率值，而是一个基于概率的对数计数器结果，最大为255（可表示100万次访问），超过255后如果继续频繁访问该值并不会继续增大，同时默认如果每过一分钟没有访问，该值会衰减1。 取值范围： 不涉及。

状态码：400

表 5-560 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见错误码。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-561 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-562 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-563 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-564 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/hotkey-task
```

响应示例

状态码：200

创建热key分析任务成功。

```
{
  "id" : "aa735b6f-3d70-44e3-9771-66c6fae7459a",
  "instance_id" : "5560df16-cebf-4473-95c4-d1b573c16e79",
  "status" : "waiting",
  "scan_type" : "manual",
  "created_at" : "2020-06-15T02:21:18.669Z",
  "num" : 0
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateHotkeyScanTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateHotkeyScanTaskRequest request = new CreateHotkeyScanTaskRequest();
        request.withInstanceId("{instance_id}");
        try {
            CreateHotkeyScanTaskResponse response = client.createHotkeyScanTask(request);
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateHotkeyScanTaskRequest()
        request.instance_id = "{instance_id}"
        response = client.create_hotkey_scan_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateHotkeyScanTaskRequest{}
request.InstanceId = "{instance_id}"
response, err := client.CreateHotkeyScanTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建热key分析任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.10 查询热 key 分析任务列表

功能介绍

查询热key分析历史记录。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/hotkey-tasks

表 5-565 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-566 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10
status	否	String	参数解释： 过期key扫描任务的状态。 约束限制： 不涉及。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败。 默认取值： 不涉及。

请求参数

无

响应参数

状态码： 200

表 5-567 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
count	Integer	参数解释： 热key分析的次数。 取值范围： 不涉及。
records	Array of RecordsResponse objects	参数解释： 热key分析记录列表。 取值范围： 不涉及。

表 5-568 RecordsResponse

参数	参数类型	描述
id	String	参数解释： 分析操作的任务ID。 取值范围： 不涉及。
status	String	参数解释： 热key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。
scan_type	String	参数解释： 分析方式。 取值范围： - manual：手动分析。 - auto：自动分析。

参数	参数类型	描述
created_at	String	参数解释： 分析任务创建的时间，格式为："2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务的开始时间，格式为："2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务的结束时间，式为："2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。

状态码：400

表 5-569 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-570 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-571 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-572 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-573 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/hotkey-tasks
```

响应示例

状态码：200

查询热key分析任务列表。

```
{
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "count" : 1,
  "records" : [ {
    "id" : "858ee14c-2271-4489-8b82-7bda7459ae3e",
    "scan_type" : "manual",
    "status" : "success",
    "created_at" : "2020-06-15T02:21:18.669Z",
    "started_at" : "2020-06-15T02:21:23.534Z",
    "finished_at" : "2020-06-15T02:21:25.588Z"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListHotKeyScanTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
```

```
ListHotKeyScanTasksRequest request = new ListHotKeyScanTasksRequest();
request.withInstanceId("{instance_id}");
try {
    ListHotKeyScanTasksResponse response = client.listHotKeyScanTasks(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListHotKeyScanTasksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_hot_key_scan_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListHotKeyScanTasksRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListHotKeyScanTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询热key分析任务列表。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.11 查询热 key 分析详情

功能介绍

查询热key分析详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/hotkey-task/{hotkey_id}

表 5-574 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
hotkey_id	是	String	参数解释： 热key分析任务ID。可调用 查询热key分析任务列表 接口，通过响应参数id的返回值获取。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-575 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 热key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 热key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。

参数	参数类型	描述
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
num	Integer	参数解释： 热key的数量。 取值范围： 不涉及。
keys	Array of HotkeysBody objects	参数解释： 热key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-576 HotkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： - string：字符串类型。 - list：列表类型。 - set：集合类型。 - zset：有序类型。 - hash：哈希类型。
shard	String	参数解释： 热key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 热key所在的DB。 取值范围： 0-255
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： - count：key的数量。 - byte：key的大小。

参数	参数类型	描述
freq	Integer	参数解释： 表示某个key在一段时间的访问频度，会随着访问的频率而变化。 该值并不是简单的访问频率值，而是一个基于概率的对数计数器结果，最大为255（可表示100万次访问），超过255后如果继续频繁访问该值并不会继续增大，同时默认如果每过一分钟没有访问，该值会衰减1。 取值范围： 不涉及。

状态码：400

表 5-577 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见错误码。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-578 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-579 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-580 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-581 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/hotkey-task/0ccb25d5-27cf-4188-b5ea-987730a85371
```

响应示例

状态码：200

查询热key分析详情成功。

```
{
  "id" : "858ee14c-2271-4489-8b82-7bda7459ae3e",
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "status" : "success",
  "scan_type" : "manual",
  "created_at" : "2020-06-15T02:21:18.669Z",
  "started_at" : "2020-06-15T02:21:23.534Z",
  "finished_at" : "2020-06-15T02:21:25.588Z",
  "keys" : [ {
    "name" : "dcs-hotkey-test",
    "type" : "string",
    "shard" : "192.168.1.134:6379",
    "db" : 0,
    "size" : 3000,
    "unit" : "byte",
    "freq" : 4
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowHotkeyTaskDetailsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
        .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowHotkeyTaskDetailsRequest request = new ShowHotkeyTaskDetailsRequest();
request.withInstanceId("{instance_id}");
request.withHotkeyId("{hotkey_id}");
try {
    ShowHotkeyTaskDetailsResponse response = client.showHotkeyTaskDetails(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowHotkeyTaskDetailsRequest()
        request.instance_id = "{instance_id}"
        request.hotkey_id = "{hotkey_id}"
        response = client.show_hotkey_task_details(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowHotkeyTaskDetailsRequest{}
    request.InstanceId = "{instance_id}"
    request.HotkeyId = "{hotkey_id}"
    response, err := client.ShowHotkeyTaskDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询热key分析详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.12 删除热 key 分析任务

功能介绍

删除热key分析任务。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/hotkey-task/{hotkey_id}

表 5-582 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
hotkey_id	是	String	参数解释： 热key分析任务ID。可调用 查询热key分析任务列表 接口，通过响应参数id的返回值获取。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-583 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 热key分析任务ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 热key分析任务的状态。 取值范围： - waiting：待处理。 - running：处理中。 - success：成功。 - failed：失败。

参数	参数类型	描述
scan_type	String	参数解释： 分析方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 分析任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。
num	Integer	参数解释： 热key的数量。 取值范围： 不涉及。
keys	Array of HotkeysBody objects	参数解释： 热key记录（创建分析任务时此值为null，不返回）。 取值范围： 不涉及。

表 5-584 HotkeysBody

参数	参数类型	描述
name	String	参数解释： Key名称。 取值范围： 不涉及。
type	String	参数解释： Key类型。 取值范围： • string：字符串类型。 • list：列表类型。 • set：集合类型。 • zset：有序类型。 • hash：哈希类型。
shard	String	参数解释： 热key所在的分片，仅在实例类型为集群时支持，格式为ip:port。 取值范围： 不涉及。
db	Integer	参数解释： 热key所在的DB。 取值范围： 0-255
size	Long	参数解释： Key的value大小。 取值范围： 不涉及。
unit	String	参数解释： 表示key的单位。 取值范围： • count：key的数量。 • byte：key的大小。

参数	参数类型	描述
freq	Integer	参数解释： 表示某个key在一段时间的访问频度，会随着访问的频率而变化。 该值并不是简单的访问频率值，而是一个基于概率的对数计数器结果，最大为255（可表示100万次访问），超过255后如果继续频繁访问该值并不会继续增大，同时默认如果每过一分钟没有访问，该值会衰减1。 取值范围： 不涉及。

状态码：204
热key分析记录不存在。

状态码：400

表 5-585 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见错误码。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-586 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-587 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-588 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-589 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/hotkey-task/0ccb25d5-27cf-4188-b5ea-987730a85371
```


响应示例

状态码：200

删除热key分析记录成功。

```
{
  "id" : "858ee14c-2271-4489-8b82-7bda7459ae3e",
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "status" : "success",
  "scan_type" : "manual",
  "created_at" : "2020-06-15T02:21:18.669Z",
  "started_at" : "2020-06-15T02:21:23.534Z",
  "finished_at" : "2020-06-15T02:21:25.588Z",
  "keys" : [ {
    "name" : "dcs-hotkey-test",
    "type" : "string",
    "shard" : "192.168.1.134:6379",
    "db" : 0,
    "size" : 3000,
    "unit" : "byte",
    "freq" : 4
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteHotkeyScanTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteHotkeyScanTaskRequest request = new DeleteHotkeyScanTaskRequest();
request.withInstanceId("{instance_id}");
request.withHotkeyId("{hotkey_id}");
try {
    DeleteHotkeyScanTaskResponse response = client.deleteHotkeyScanTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteHotkeyScanTaskRequest()
        request.instance_id = "{instance_id}"
        request.hotkey_id = "{hotkey_id}"
        response = client.delete_hotkey_scan_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dcs.NewDcsClient(  
        dcs.DcsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DeleteHotkeyScanTaskRequest{  
        request.InstanceId = "{instance_id}"  
        request.HotkeyId = "{hotkey_id}"  
    }  
    response, err := client.DeleteHotkeyScanTask(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除热key分析记录成功。
204	热key分析记录不存在。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.13 设置热 key 自动分析配置

功能介绍

设置热key自动分析配置。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/hotkey/autoscan

表 5-590 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-591 请求 Body 参数

参数	是否必选	参数类型	描述
enable_auto_scan	是	Boolean	参数解释： 是否开启缓存分析定时任务。 约束限制： 不涉及。 取值范围： • true • false 默认取值： 不涉及。
schedule_at	是	Array of strings	参数解释： 每日缓存分析时间，时间格式为 21:00，时间为UTC时间。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-592 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启缓存分析定时任务。 取值范围： • true • false

参数	参数类型	描述
schedule_at	Array of strings	参数解释： 每日分析时间，时间格式为21:00，时间为UTC时间。 取值范围： 不涉及。
updated_at	String	参数解释： 配置更新时间，时间格式为2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-593 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-594 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-595 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-596 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-597 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

设置热key自动分析任务，每日缓存分析时间为21：00。


```
PUT https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-
cebf-4473-95c4-d1b573c16e79/hotkey/autoscan
```

```
{
  "enable_auto_scan" : true,
  "schedule_at" : [ "21:00" ]
}
```

响应示例

状态码：200

设置热key自动分析配置成功。

```
{
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "enable_auto_scan" : true,
  "schedule_at" : [ "21:00" ],
  "updated_at" : "2020-06-17T02:42:40.793Z"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置热key自动分析任务，每日缓存分析时间为21：00。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateHotkeyAutoScanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateHotkeyAutoScanConfigRequest request = new UpdateHotkeyAutoScanConfigRequest();
request.withInstanceId("{instance_id}");
AutoscanConfigRequest body = new AutoscanConfigRequest();
List<String> listbodyScheduleAt = new ArrayList<>();
listbodyScheduleAt.add("21:00");
body.withScheduleAt(listbodyScheduleAt);
body.withEnableAutoScan(true);
request.withBody(body);
try {
    UpdateHotkeyAutoScanConfigResponse response = client.updateHotkeyAutoScanConfig(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

设置热key自动分析任务，每日缓存分析时间为21：00。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateHotkeyAutoScanConfigRequest()
        request.instance_id = "{instance_id}"
        listScheduleAtbody = [
            "21:00"
        ]
        request.body = AutoscanConfigRequest(
            schedule_at=listScheduleAtbody,
```

```
        enable_auto_scan=True
    )
    response = client.update_hotkey_auto_scan_config(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

设置热key自动分析任务，每日缓存分析时间为21:00。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateHotkeyAutoScanConfigRequest{}
    request.InstanceId = "{instance_id}"
    var listScheduleAtbody = []string{
        "21:00",
    }
    request.Body = &model.AutoscanConfigRequest{
        ScheduleAt: listScheduleAtbody,
        EnableAutoScan: true,
    }
    response, err := client.UpdateHotkeyAutoScanConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	设置热key自动分析配置成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.14 查询热 key 自动分析配置

功能介绍

查询热key自动分析配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/hotkey/autoscan

表 5-598 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-599 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启缓存分析定时任务。 取值范围： - true - false
schedule_at	Array of strings	参数解释： 每日分析时间，时间格式为21:00，时间为UTC时间。 取值范围： 不涉及。

参数	参数类型	描述
updated_at	String	参数解释： 配置更新时间，时间格式为 2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-600 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-601 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-602 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-603 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-604 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/hotkey/autoscan
```


响应示例

状态码：200

查询热key自动分析配置。

```
{
  "instance_id" : "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "enable_auto_scan" : true,
  "schedule_at" : [ "21:00" ],
  "updated_at" : "2020-06-17T02:42:40.793Z"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4922",
  "error_msg" : "Does not support hotkey analyze."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowHotkeyAutoscanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowHotkeyAutoscanConfigRequest request = new ShowHotkeyAutoscanConfigRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowHotkeyAutoscanConfigResponse response = client.showHotkeyAutoscanConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowHotkeyAutoscanConfigRequest()
        request.instance_id = "{instance_id}"
        response = client.show_hotkey_autoscan_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowHotkeyAutoscanConfigRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowHotkeyAutoscanConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询热key自动分析配置。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.15 创建过期 key 扫描任务

功能介绍

创建过期key扫描任务，Redis 3.0不支持过期key扫描。

过期key扫描会对键空间进行Redis的scan扫描，释放内存中已过期但是由于惰性删除机制而没有释放的内存空间。

过期key扫描在主节点上执行，会对实例性能有一定的影响，建议不要在业务高峰期进行。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/scan-expire-keys-task

表 5-605 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-606 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 过期key扫描记录ID。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
status	String	参数解释： 过期key扫描任务的状态。 取值范围： • waiting：待处理。 • running：处理中。 • success：成功。 • failed：失败
scan_type	String	参数解释： 扫描方式。 取值范围： • manual：手动分析。 • auto：自动分析。
created_at	String	参数解释： 扫描任务创建时间，格式为： "2020-06-15T02:21:18.669Z"。 取值范围： 不涉及。
started_at	String	参数解释： 扫描任务开始时间，格式为： "2020-06-15T02:21:18.669Z"（创建扫描任务时此值为null，不返回）。 取值范围： 不涉及。
finished_at	String	参数解释： 扫描任务结束时间，格式为： "2020-06-15T02:21:18.669Z"（创建扫描任务时此值为null，不返回）。 取值范围： 不涉及。

状态码：400

表 5-607 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-608 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-609 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-610 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-611 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/scan-expire-keys-task
```

响应示例

状态码：200

创建过期key扫描任务成功。

```
{
  "id": "aa735b6f-3d70-44e3-9771-66c6fae7459a",
  "instance_id": "5560df16-cebf-4473-95c4-d1b573c16e79",
  "status": "waiting",
  "scan_type": "manual",
  "created_at": "2020-06-15T02:21:18.669Z"
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4818",
  "error_msg": "Do not support auto expire scan."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateAutoExpireScanTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        CreateAutoExpireScanTaskRequest request = new CreateAutoExpireScanTaskRequest();
        request.withInstanceId("{instance_id}");
        try {
            CreateAutoExpireScanTaskResponse response = client.createAutoExpireScanTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateAutoExpireScanTaskRequest()
    request.instance_id = "{instance_id}"
    response = client.create_auto_expire_scan_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAutoExpireScanTaskRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.CreateAutoExpireScanTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建过期key扫描任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.16 查询自动扫描配置

功能介绍

查询过期key自动扫描配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/scan-expire-keys/autoscan-config

表 5-612 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-613 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启自动扫描。 取值范围： - true - false
first_scan_at	String	参数解释： 首次扫描时间，例如： 2023-07-07T15:00:05.000。 取值范围： 不涉及。
interval	Integer	参数解释： 扫描间隔时间（单位：秒）。 取值范围： 不涉及。

参数	参数类型	描述
timeout	Integer	参数解释： 扫描超时时间（单位：秒）。 取值范围： 不涉及。
scan_keys_count	Integer	参数解释： 迭代扫描key数量。 取值范围： 不涉及。
updated_at	AnyType	参数解释： 配置更新时间，时间格式为 2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-614 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-615 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-616 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-617 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-618 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/scan-expire-keys/autoscan-config
```

响应示例

状态码：200

查询自动扫描配置成功。

```
{
  "instance_id" : "162d315d-a67d-4ba1-8ca5-8d0e95c5bf7c",
  "enable_auto_scan" : false,
  "first_scan_at" : null,
  "interval" : null,
  "timeout" : null,
  "scan_keys_count" : null,
  "updated_at" : null
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowExpireAutoScanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowExpireAutoScanConfigRequest request = new ShowExpireAutoScanConfigRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowExpireAutoScanConfigResponse response = client.showExpireAutoScanConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
        }
    }
}
```



```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowExpireAutoScanConfigRequest()
        request.instance_id = "{instance_id}"
        response = client.show_expire_auto_scan_config(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```
WithSk(sk).
WithProjectId(projectId).
Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowExpireAutoScanConfigRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowExpireAutoScanConfig(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询自动扫描配置成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.17 修改自动扫描配置

功能介绍

修改过期key自动扫描配置。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/scan-expire-keys/autoscan-config

表 5-619 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-620 请求 Body 参数

参数	是否必选	参数类型	描述
enable_auto_scan	否	Boolean	参数解释： 是否开启缓存分析定时任务。 约束限制： 不涉及。 取值范围： • true • false 默认取值： 不涉及。

参数	是否必选	参数类型	描述
first_scan_at	否	String	参数解释： 首次扫描时间，例如： 2023-07-07T15:00:05.000。当 enable_auto_scan设置为true 时，该参数为必选。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
interval	否	Integer	参数解释： 扫描间隔时间（单位：秒）。当 enable_auto_scan设置为true 时，该参数为必选。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
timeout	否	Integer	参数解释： 扫描超时时间（单位：秒）。扫 描超过设置的超时时间后，返回 失败，以便能继续进行下一轮扫 描。当enable_auto_scan设置为 true时，该参数为必选。 约束限制： 扫描超时时间必须大于等于扫描 间隔时间的2倍。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
scan_keys_count	否	Integer	参数解释： 迭代扫描key的数量。例如： redis中有1000万个key，迭代扫描key数量设为1000，则迭代1万次可完成全库扫描。当enable_auto_scan设置为true时，该参数为必选。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-621 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_auto_scan	Boolean	参数解释： 是否开启自动扫描。 取值范围： - true - false
first_scan_at	String	参数解释： 首次扫描时间，例如： 2023-07-07T15:00:05.000。 取值范围： 不涉及。
interval	Integer	参数解释： 扫描间隔时间（单位：秒）。 取值范围： 不涉及。

参数	参数类型	描述
timeout	Integer	参数解释： 扫描超时时间（单位：秒）。 取值范围： 不涉及。
scan_keys_count	Integer	参数解释： 迭代扫描key数量。 取值范围： 不涉及。
updated_at	AnyType	参数解释： 配置更新时间，时间格式为 2020-06-15T02:21:18.669Z。 取值范围： 不涉及。

状态码：400

表 5-622 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-623 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-624 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-625 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-626 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/scan-expire-keys/autoscan-config
{
```



```
"enable_auto_scan": "true",
"first_scan_at": "2023-02-07T15:00:05.000Z",
"interval": 86400,
"timeout": 172800,
"scan_keys_count": 10
}
```

响应示例

状态码：200

修改自动扫描配置成功。

```
{
  "instance_id": "162d315d-a67d-4ba1-8ca5-8d0e95c5bf7c",
  "enable_auto_scan": true,
  "first_scan_at": "2023-06-13T21:00:00.000Z",
  "interval": 86400,
  "timeout": 172800,
  "scan_keys_count": 10,
  "updated_at": "2023-06-13T13:13:12.685Z"
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4822",
  "error_msg": "The scan_keys_count parameter is invalid. Its must Greater than or equal to 10, and Less than or equal to 1000."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateExpireAutoScanConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateExpireAutoScanConfigRequest request = new UpdateExpireAutoScanConfigRequest();
request.withInstanceId("{instance_id}");
UpdateAutoScanConfigRequestBody body = new UpdateAutoScanConfigRequestBody();
body.withScanKeysCount(10);
body.withTimeout(172800);
body.withInterval(86400);
body.withFirstScanAt("2023-02-07T15:00:05.000Z");
body.withEnableAutoScan(true);
request.withBody(body);
try {
    UpdateExpireAutoScanConfigResponse response = client.updateExpireAutoScanConfig(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateExpireAutoScanConfigRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateAutoScanConfigRequestBody(
            scan_keys_count=10,
            timeout=172800,
            interval=86400,
            first_scan_at="2023-02-07T15:00:05.000Z",
            enable_auto_scan=True
        )
        response = client.update_expire_auto_scan_config(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateExpireAutoScanConfigRequest{}
    request.InstanceId = "{instance_id}"
    scanKeysCountUpdateAutoScanConfigRequestBody := int32(10)
    timeoutUpdateAutoScanConfigRequestBody := int32(172800)
    intervalUpdateAutoScanConfigRequestBody := int32(86400)
    firstScanAtUpdateAutoScanConfigRequestBody := "2023-02-07T15:00:05.000Z"
    enableAutoScanUpdateAutoScanConfigRequestBody := true
    request.Body = &model.UpdateAutoScanConfigRequestBody{
        ScanKeysCount: &scanKeysCountUpdateAutoScanConfigRequestBody,
        Timeout: &timeoutUpdateAutoScanConfigRequestBody,
        Interval: &intervalUpdateAutoScanConfigRequestBody,
        FirstScanAt: &firstScanAtUpdateAutoScanConfigRequestBody,
        EnableAutoScan: &enableAutoScanUpdateAutoScanConfigRequestBody,
    }
    response, err := client.UpdateExpireAutoScanConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	修改自动扫描配置成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.8.18 下载热 key

功能介绍

下载热key。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/hotkey/{task_id}/export

表 5-627 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 热Key分析任务ID，可通过DCS控制台进入实例缓存分析界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-628 响应 Body 参数

参数	参数类型	描述
-	File	下载热Key的文件，提供实例热key的大小，出现的频率和所在的数据库。

状态码：400

表 5-629 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-630 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/hotkey/{task_id}/export
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DownloadHotKeySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DownloadHotKeyRequest request = new DownloadHotKeyRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            DownloadHotKeyResponse response = client.downloadHotKey(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DownloadHotKeyRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.download_hot_key(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.DownloadHotKeyRequest{}
request.InstanceId = "{instance_id}"
request.TaskId = "{task_id}"
response, err := client.DownloadHotKey(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下载热Key成功。
400	非法请求。
500	内部服务器错误。

错误码

请参见[错误码](#)。

5.9 日志管理

5.9.1 查询慢日志

功能介绍

查询慢日志。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/slowlog

表 5-631 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-632 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示从此偏移量开始查询。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10
sort_key	否	String	参数解释： 返回结果按该关键字排序。 约束限制： 不涉及。 取值范围： - start_time：慢日志的开始时间。 - duration：慢日志的耗时长短。 默认取值： “start_time”
sort_dir	否	String	参数解释： 返回结果按照降序或升序排序。 约束限制： 不涉及。 取值范围： - desc：降序。 - asc：升序。 默认取值： “desc”
start_time	是	String	参数解释： 查询开始时间，时间为UTC时间的Unix时间戳。如： 1598803200000。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
end_time	是	String	参数解释： 查询结束时间，时间为UTC时间的Unix时间戳。如：1599494399000。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-633 响应 Body 参数

参数	参数类型	描述
total_num	Integer	参数解释： 慢日志总数。 取值范围： 不涉及。
count	Integer	参数解释： 慢日志总数。 取值范围： 不涉及。
slowlogs	Array of SlowlogItem objects	参数解释： 慢日志列表。 取值范围： 不涉及。

表 5-634 SlowlogItem

参数	参数类型	描述
id	Integer	参数解释： 慢日志的唯一标识。 取值范围： 不涉及。
command	String	参数解释： 慢命令。 取值范围： 不涉及。
start_time	String	参数解释： 执行开始时间，格式为 “2020-06-19T07:06:07Z” 。 取值范围： 不涉及。
duration	String	参数解释： 持续时间，单位是μs。 取值范围： 不涉及。
shard_name	String	参数解释： 慢命令所在的分片名称，仅在实例类型为集群时支持。 取值范围： 不涉及。
database_id	Integer	参数解释： 数据库ID，当前只对指定客户开放。 取值范围： 0-255。
username	String	参数解释： 操作慢日志的账号名称，当前只对指定客户开放。 取值范围： 不涉及。
node_role	String	参数解释： 节点类型。 取值范围： 不涉及。

参数	参数类型	描述
client_ip	String	参数解释： 客户端IP地址。 取值范围： 不涉及。

状态码：400

表 5-635 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-636 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-637 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-638 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-639 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-  
cebf-4473-95c4-d1b573c16e79/slowlog?start_time=1598803200000&end_time=1599494399000
```

响应示例

状态码：200

查询慢日志成功。

```
{  
  "total_num" : 1,
```



```
"count" : 1,
"slowlogs" : [ {
  "id" : 1,
  "start_time" : "2020-06-19T07:06:07Z",
  "command" : "SLOWLOG GET 128",
  "duration" : "17",
  "shard_name" : "group-0",
  "database_id" : 1,
  "username" : "default",
  "node_role" : "master",
  "client_ip" : "127.0.0.1:6379"
} ]
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4800",
  "error_msg" : "Invalid start_time in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListSlowlogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSlowlogRequest request = new ListSlowlogRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListSlowlogResponse response = client.listSlowlog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSlowlogRequest()
        request.instance_id = "{instance_id}"
        response = client.list_slowlog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListSlowlogRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListSlowlog(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询慢日志成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.9.2 查询 Redis 运行日志列表

功能介绍

查询Redis运行日志列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/redislog

表 5-640 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-641 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10
log_type	是	String	参数解释： 返回日志的类型。 约束限制： 不涉及。 取值范围： run：Redis运行日志。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-642 响应 Body 参数

参数	参数类型	描述
total_num	Integer	参数解释： 日志总数。 取值范围： 不涉及。
file_list	Array of RunlogItem objects	参数解释： 运行日志列表。

表 5-643 RunlogItem

参数	参数类型	描述
id	String	参数解释： 日志的唯一标识。 取值范围： 不涉及。
file_name	String	参数解释： 运行日志文件名。 取值范围： 不涉及。
group_name	String	参数解释： 分片名称。 取值范围： 不涉及。
replication_ip	String	参数解释： 采集运行日志所在副本的IP。 取值范围： 不涉及。
status	String	参数解释： 获取运行日志状态。 取值范围： 不涉及。
time	String	参数解释： 运行日志采集的日期，格式为"yyyy-MM-dd"。 取值范围： 不涉及。

参数	参数类型	描述
backup_id	String	参数解释： 日志文件的ID。 取值范围： 不涉及。

状态码：204
查询成功，无Redislog。
状态码：400

表 5-644 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-645 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-646 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-647 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-648 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/redislog?log_type=run&offset=1&limit=10&replication_id=50fa23b4-74d3-4d5c-98b9-39188a1f2daf
```

响应示例

状态码：200

参数解释：

查询运行日志成功。如果没有日志则会返回'204'。

```
{
  "total_num" : 1,
  "file_list" : [ {
    "id" : "e79f2d97-e972-41f2-b51c-d18f65f39a45",
    "file_name" : "redis_192.168.0.142_2021-04-16.log",
    "status" : "succeed",
    "time" : "2021-04-16",
    "replication_ip" : "192.168.0.142",
    "group_name" : "group-0",
    "backup_id" : "e79f2d97-e972-41f2-b51c-d18f65f39a45"
  } ]
}
```

状态码：400

参数解释：

非法请求。

```
{
  "error_code" : "DCS.4800",
  "error_msg" : "Invalid logType."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListRedislogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ListRedislogRequest request = new ListRedislogRequest();
request.withInstanceId("{instance_id}");
try {
    ListRedislogResponse response = client.listRedislog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListRedislogRequest()
        request.instance_id = "{instance_id}"
        response = client.list_redislog(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRedislogRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListRedislog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询运行日志成功。如果没有日志则会返回'204'。
204	查询成功，无Redislog。
400	非法请求。
401	认证失败。
403	请求被拒绝。
404	路径错误。
500	内部错误。

错误码

请参见[错误码](#)。

5.9.3 采集 Redis 运行日志

功能介绍

采集Redis运行日志。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/redislog

表 5-649 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

表 5-650 Query 参数

参数	是否必选	参数类型	描述
query_time	否	Integer	日期偏移量，表示从过去的n天开始查询，默认为0。取值范围：0，1，3，7。取值说明：传入0则表示查询今天的日志，传入7则表示查询过去7天的日志。
log_type	是	String	返回日志的类型，当前仅支持Redis运行日志，类型为run。
replication_id	否	String	副本ID，可以从分片与副本中查询对应节点的副本ID，当实例不为单机实例时，该参数为必填。

请求参数

无

响应参数

状态码：204

采集运行日志成功。

状态码：400

表 5-651 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-652 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-653 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-654 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-655 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-
cebf-4473-95c4-d1b573c16e79/redislog?
log_type=run&query_time=7&replication_id=50fa23b4-74d3-4d5c-98b9-39188a1f2daf
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4800",
  "error_msg" : "Invalid logType."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateRedislogSolution {
```



```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DcsClient client = DcsClient.newBuilder()
        .withCredential(auth)
        .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateRedislogRequest request = new CreateRedislogRequest();
    request.withInstanceId("{instance_id}");
    try {
        CreateRedislogResponse response = client.createRedislog(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateRedislogRequest()
        request.instance_id = "{instance_id}"
```

```
response = client.create_redislog(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateRedislogRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.CreateRedislog(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	采集运行日志成功。
400	非法请求。

状态码	描述
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.9.4 获取日志下载链接

功能介绍

获取日志下载链接。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/redislog/{id}/links

表 5-656 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
id	是	String	日志的唯一ID，来自于查询运行日志查询接口。

请求参数

无

响应参数

状态码：200

表 5-657 响应 Body 参数

参数	参数类型	描述
id	String	日志ID。
backup_id	String	后台任务ID。
link	String	日志下载链接，默认有效时间为24小时。

状态码：400

表 5-658 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-659 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-660 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-661 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-662 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/0524e839a54847039e9b2b46e8aa788a/instances/5c597ce5-9823-48aa-bf4d-2defb9385b4a/redislog/3ea6ce21-048e-447a-83c3-3fb004b88439/links
```

响应示例

状态码：200

获取日志下载链接成功。

```
{
  "id" : "3ea6ce21-048e-447a-83c3-3fb004b88439",
  "backup_id" : "3ea6ce21-048e-447a-83c3-3fb004b88439",
  "link" : "https://bucketxxxxx.{obs_endpoint}:443/xxxxx/redis_192.168.x.x_2021-04-16.log?AWSAccessKeyId=xxxxx"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateRedislogDownloadLinkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateRedislogDownloadLinkRequest request = new CreateRedislogDownloadLinkRequest();
        request.withInstanceId("{instance_id}");
        request.withId("{id}");
        try {
            CreateRedislogDownloadLinkResponse response = client.createRedislogDownloadLink(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateRedislogDownloadLinkRequest()
        request.instance_id = "{instance_id}"
        request.id = "{id}"
        response = client.create_redislog_download_link(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```



```
WithProjectId(projectId).
Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateRedislogDownloadLinkRequest{}
request.InstanceId = "{instance_id}"
request.Id = "{id}"
response, err := client.CreateRedislogDownloadLink(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	获取日志下载链接成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.10 账号管理

5.10.1 创建 ACL 账号

功能介绍

为redis4.0/5.0实例创建权限访问账号，包含读写和只读权限。

如果实例默认账号已开启免密访问，您创建的普通账号不能使用，如需使用普通账号请先关闭默认账号的免密访问。

单机、主备实例默认账号的密码不能带有冒号(:)，否则无法创建普通账号。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/accounts

表 5-663 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-664 请求 Body 参数

参数	是否必选	参数类型	描述
account_name	是	String	参数解释： ACL账号名称。 约束限制： - 以字母开头。 - 内容由字母、数字、中划线、下划线组成。 - 长度范围[1~64]个字符。 取值范围： 不涉及。 默认取值： 不涉及。
account_role	是	String	参数解释： ACL账号权限。 约束限制： 不涉及。 取值范围： - read：只读账号 - write：读写账号 默认取值： 不涉及。

参数	是否必选	参数类型	描述
account_password	是	String	参数解释： ACL账号密码。 约束限制： • 输入长度为8到64位的字符串。 • 不能使用正序或逆序用户名作为密码。 • 必须包含如下四种字符中的三种组合（不允许包含冒号）： - 小写字母 - 大写字母 - 数字 - 特殊字符包括（`~!@#\$%^&*()-_+=\ {}";<.>/?`） 取值范围： 不涉及。 默认取值： 不涉及。
description	否	String	参数解释： ACL账号描述。 约束限制： 长度不超过1024的字符串。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
account_read_policy	否	String	参数解释： 指定账号的读请求转发到主节点或备节点。 约束限制： - 仅proxy集群和读写分离实例支持，proxy集群未开启读写分离不支持设置。 - 读写分离配置参考“用户指南>管理实例>修改DCS实例配置参数”中backend-master-only参数说明。 取值范围： - master：读请求转发到主节点。 - replica：读请求转发到备节点。 - master-replica：读请求转发到主备节点。 默认取值： 默认不设置。不设置时，读请求按照实例读写分离配置进行转发。 - 完全读写分离配置参考“用户指南>管理实例>修改DCS实例配置参数”中read-only-slave-when-wr-split参数说明。 - proxy集群实例 - 默认未开启读写分离，读请求转发到主节点。 - 开启读写分离 - 默认开启完全读写分离，读请求转发到备节点。 - 未开启完全读写分离，读请求随机转发到主备节点。 - 读写分离实例 - 默认支持读写分离 - 默认开启完全读写分离，读请求转发到备节点。

参数	是否必选	参数类型	描述
			未开启完全读写分离，读请求随机转发到主备节点。

响应参数

状态码：200

参数解释：

ACL账号创建任务下发成功。

状态码：400

表 5-665 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-666 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-667 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-668 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-669 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

创建ACL只读权限账号，账号名为user。


```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts

{
  "account_name": "user",
  "account_role": "read",
  "account_password": "xxxxx"
}
```

响应示例

状态码：400

参数解释：

非法请求。

```
{
  "error_code": "DCS.4961",
  "error_msg": "Acl account request param invalid."
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建ACL只读权限账号，账号名为user。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateAclAccountSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateAclAccountRequest request = new CreateAclAccountRequest();
        request.withInstanceId("{instance_id}");
        CreateAclAccountRequestBody body = new CreateAclAccountRequestBody();
        body.withAccountPassword("xxxxx");
```

```
body.withAccountRole(CreateAclAccountRequestBody.AccountRoleEnum.fromValue("read"));
body.withAccountName("user");
request.withBody(body);
try {
    CreateAclAccountResponse response = client.createAclAccount(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建ACL只读权限账号，账号名为user。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateAclAccountRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateAclAccountRequestBody(
            account_password="xxxxx",
            account_role="read",
            account_name="user"
        )
        response = client.create_acl_account(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建ACL只读权限账号，账号名为user。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAclAccountRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.CreateAclAccountRequestBody{
        AccountPassword: "xxxxx",
        AccountRole: model.GetCreateAclAccountRequestBodyAccountRoleEnum().READ,
        AccountName: "user",
    }
    response, err := client.CreateAclAccount(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： ACL账号创建任务下发成功。
400	参数解释： 非法请求。

状态码	描述
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.10.2 查询 ACL 账户列表

功能介绍

查询ACL账户列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/accounts

表 5-670 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-671 响应 Body 参数

参数	参数类型	描述
accounts	Array of AclAccountResp objects	参数解释： ACL账号列表。

表 5-672 AclAccountResp

参数	参数类型	描述
account_id	String	参数解释： ACL账号ID。 取值范围： 不涉及。
account_name	String	参数解释： ACL账号名。 取值范围： 不涉及。

参数	参数类型	描述
account_type	String	参数解释： ACL账号类型。 取值范围： normal：普通账号。 default：默认账号。
instance_id	String	参数解释： 账号所属实例ID。 取值范围： 不涉及。
status	String	参数解释： ACL账号状态。 取值范围： • CREATING：账号创建中。 • AVAILABLE：账号可用。 • DELETEFAILED：账号删除失败。 • DELETING：账号删除中。 • UPDATING：账号更新中。 • ERROR：账号异常。
account_role	String	参数解释： 账号权限。 取值范围： read：只读账号。 write：读写账号。
description	String	参数解释： 账号描述。 取值范围： 不涉及。
error_code	String	参数解释： 错误码（该参数暂未使用，赋值为null）。 取值范围： 不涉及。

参数	参数类型	描述
account_read_policy	String	参数解释： 指定账号的读请求转发到主节点或备节点。 取值范围： • master：读请求转发到主节点。 • replica：读请求转发到备节点。 • master-replica：读请求转发到主备节点。

状态码：400

表 5-673 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-674 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-675 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-676 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-677 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts
```

响应示例

状态码：200

参数解释：

ACL账户列表。

```
{
  "accounts": [ {
    "account_id": "cb067c67-f08b-4024-af14-f78b58b76a44",
    "account_name": "roUser",
    "account_type": "normal",
    "instance_id": "10fda556-51c3-4cfa-ab6f-37667428da6e",
    "status": "available",
    "account_role": "read",
    "description": "这是一个只读账号",
    "error_code": null
  } ]
}
```

状态码：400

参数解释：

非法请求。

```
{
  "error_code": "DCS.4964",
  "error_msg": "Failed to query acl list."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListAclAccountsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ListAclAccountsRequest request = new ListAclAccountsRequest();
request.withInstanceId("{instance_id}");
try {
    ListAclAccountsResponse response = client.listAclAccounts(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAclAccountsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_acl_accounts(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAclAccountsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListAclAccounts(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： ACL账户列表。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。

状态码	描述
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.10.3 修改 ACL 账号密码

功能介绍

修改ACL账号密码。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/accounts/{account_id}/password/
modify

表 5-678 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
account_id	是	String	ACL账号ID。

请求参数

表 5-679 请求 Body 参数

参数	是否必选	参数类型	描述
old_password	否	String	旧密码。

参数	是否必选	参数类型	描述
new_password	否	String	新密码。 • 输入长度为8到64位的字符串。 • 不能包含正序逆序用户名。 • 必须包含如下四种字符中的三种组合（不允许包含冒号）： - 小写字母 - 大写字母 - 数字 - 特殊字符包括（`~!@#\$%^&*()-_+= {>,.<./?`）

响应参数

状态码：200

表 5-680 响应 Body 参数

参数	参数类型	描述
lock_time	String	锁定时间。验证密码失败时和锁定时该参数返回不为null。
result	String	密码修改结果： • 成功：success； • 密码验证失败：passwordFailed； • 已锁定：locked； • 失败：failed。
lock_time_left	String	锁定剩余时间。锁定时该参数返回不为null。
retry_times_left	String	密码验证剩余次数。验证密码失败时该参数返回不为null。
message	String	修改结果说明信息。

状态码：400

表 5-681 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-682 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-683 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-684 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-685 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts/{account_id}/password/modify
{
  "old_password": "xxxxxx",
  "new_password": "xxxxxx"
}
```

响应示例

状态码：200

修改密码成功。

```
{
  "lock_time": "0",
  "result": "success",
  "lock_time_left": "0",
  "retry_times_left": "5",
  "message": "Modify DCSInstance password success."
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4961",
  "error_msg": "Acl account request param invalid."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateAclAccountPassWordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAclAccountPassWordRequest request = new UpdateAclAccountPassWordRequest();
        request.withInstanceId("{instance_id}");
        request.withAccountId("{account_id}");
        AclAccountModifyPasswordBody body = new AclAccountModifyPasswordBody();
        body.withNewPassword("xxxxxx");
        body.withOldPassword("xxxxxx");
        request.withBody(body);
        try {
            UpdateAclAccountPassWordResponse response = client.updateAclAccountPassWord(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAclAccountPassWordRequest()
        request.instance_id = "{instance_id}"
        request.account_id = "{account_id}"
        request.body = AclAccountModifyPasswordBody(
            new_password="xxxxxx",
            old_password="xxxxxx"
        )
        response = client.update_acl_account_pass_word(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAclAccountPassWordRequest{
        request.InstanceId = "{instance_id}"
```

```
request.AccountId = "{account_id}"
newPasswordAclAccountModifyPasswordBody:= "xxxxxx"
oldPasswordAclAccountModifyPasswordBody:= "xxxxxx"
request.Body = &model.AclAccountModifyPasswordBody{
    NewPassword: &newPasswordAclAccountModifyPasswordBody,
    OldPassword: &oldPasswordAclAccountModifyPasswordBody,
}
response, err := client.UpdateAclAccountPassWord(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	修改密码成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.10.4 重置 ACL 账号密码

功能介绍

重置ACL账号密码。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/accounts/{account_id}/password/reset

表 5-686 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
account_id	是	String	ACL账号ID。

请求参数

表 5-687 请求 Body 参数

参数	是否必选	参数类型	描述
old_password	否	String	旧密码。
new_password	否	String	新密码。 • 输入长度为8到64位的字符串。 • 不能包含正序逆序用户名。 • 必须包含如下四种字符中的三种组合（不允许包含冒号）： - 小写字母 - 大写字母 - 数字 - 特殊字符包括（`~!@#\$%^&*()-_+= {>,.<./?`）

响应参数

状态码：200

表 5-688 响应 Body 参数

参数	参数类型	描述
code	String	任务执行成功代码。
message	String	重置结果说明信息。
lock_time	String	锁定时间。验证失败时和锁定时该参数返回不为null。
lock_time_left	String	锁定剩余时间。锁定时该参数返回不为null。

参数	参数类型	描述
retry_times_left	String	密码验证剩余次数。验证失败时该参数返回不为null。

状态码：400

表 5-689 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-690 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-691 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-692 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-693 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts/{account_id}/password/reset"
{
  "old_password" : "xxxxxx",
  "new_password" : "xxxxxx"
}
```

响应示例

状态码：200

重置密码成功。

```
{
  "code" : "1",
  "message" : "Reset DCSInstance password success.",
  "lock_time" : "0",
  "lock_time_left" : "0",
  "retry_times_left" : "5"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4961"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ResetAclAccountPassWordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetAclAccountPassWordRequest request = new ResetAclAccountPassWordRequest();
        request.withInstanceId("{instance_id}");
        request.withAccountId("{account_id}");
        AclAccountResetPasswordBody body = new AclAccountResetPasswordBody();
        body.withNewPassword("xxxxxx");
        body.withOldPassword("xxxxxx");
        request.withBody(body);
        try {
            ResetAclAccountPassWordResponse response = client.resetAclAccountPassWord(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetAclAccountPassWordRequest()
        request.instance_id = "{instance_id}"
        request.account_id = "{account_id}"
        request.body = AclAccountResetPasswordBody(
            new_password="xxxxxx",
            old_password="xxxxxx"
        )
        response = client.reset_acl_account_pass_word(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResetAclAccountPassWordRequest{
    request.InstanceId = "{instance_id}"
    request.AccountId = "{account_id}"
    newPasswordAclAccountResetPasswordBody:= "xxxxxx"
    oldPasswordAclAccountResetPasswordBody:= "xxxxxx"
    request.Body = &model.AclAccountResetPasswordBody{
        NewPassword: &newPasswordAclAccountResetPasswordBody,
        OldPassword: &oldPasswordAclAccountResetPasswordBody,
    }
}
response, err := client.ResetAclAccountPassWord(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	重置密码成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.10.5 修改 ACL 角色

功能介绍

修改ACL账号的类型。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/accounts/{account_id}/role

表 5-694 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
account_id	是	String	ACL账号ID。

请求参数

表 5-695 请求 Body 参数

参数	是否必选	参数类型	描述
account_role	否	String	账号权限，read：只读账号，write：读写账号。

参数	是否必选	参数类型	描述
account_read_policy	否	String	参数解释： 指定账号的读请求转发到主节点或备节点。 约束限制： • 仅proxy集群和读写分离实例支持，proxy集群未开启读写分离不支持设置。 • 读写分离配置参考“用户指南>管理实例>修改DCS实例配置参数”中backend-master-only参数说明。 • master：读请求转发到主节点。 • replica：读请求转发到备节点。 • master-replica：读请求转发到主备节点。 • 完全读写分离配置参考“用户指南>管理实例>修改DCS实例配置参数”中read-only-slave-when-wr-split参数说明。 • proxy集群实例 - 默认未开启读写分离，读请求转发到主节点。 - 开启读写分离 - 默认开启完全读写分离，读请求转发到备节点。 - 未开启完全读写分离，读请求随机转发到主备节点。 • 读写分离实例 - 默认支持读写分离 - 默认开启完全读写分离，读请求转发到备节点。 - 未开启完全读写分离，读请求随机转发到主备节点。 取值范围： 默认取值：

参数	是否必选	参数类型	描述
			默认不设置。不设置时，读请求按照实例读写分离配置进行转发。

响应参数

状态码：200

表 5-696 响应 Body 参数

参数	参数类型	描述
instance_id	String	账号所属实例ID。
job_id	String	修改ACL账号访问权限JOB的ID。

状态码：400

表 5-697 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-698 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-699 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-700 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-701 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts/{account_id}/role
{
```



```
"account_role" : "read"
}
```

响应示例

状态码：200

重置密码成功。

```
{
  "job_id" : "ff80808284d4c6d50184e8cff10248bd",
  "instance_id" : "10fda556-51c3-4cfa-ab6f-37667428da6e"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4961",
  "error_msg" : "Acl account request param invalid."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateAclAccountSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();

        UpdateAclAccountRequest request = new UpdateAclAccountRequest();
        request.withInstanceId("{instance_id}");
        request.withAccountId("{account_id}");
        AclAccountRoleModifyBody body = new AclAccountRoleModifyBody();
        body.withAccountRole(AclAccountRoleModifyBody.AccountRoleEnum.fromValue("read"));
    }
}
```

```
request.withBody(body);
try {
    UpdateAclAccountResponse response = client.updateAclAccount(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAclAccountRequest()
        request.instance_id = "{instance_id}"
        request.account_id = "{account_id}"
        request.body = AclAccountRoleModifyBody(
            account_role="read"
        )
        response = client.update_acl_account(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAclAccountRequest{}
    request.InstanceId = "{instance_id}"
    request.AccountId = "{account_id}"
    accountRoleAclAccountRoleModifyBody:= model.GetAclAccountRoleModifyBodyAccountRoleEnum().READ
    request.Body = &model.AclAccountRoleModifyBody{
        AccountRole: &accountRoleAclAccountRoleModifyBody,
    }
    response, err := client.UpdateAclAccount(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	重置密码成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.10.6 删除 ACL 账号

功能介绍

删除所创建的ACL普通账号。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/accounts/{account_id}

表 5-702 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
account_id	是	String	参数解释： ACL账号ID。请调用 查询ACL账户列表 接口获取。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200
ACL账号删除任务下发成功。

状态码：400

表 5-703 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-704 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-705 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-706 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-707 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts/{account_id}
```

响应示例

状态码：400

参数解释：

非法请求。

```
{
  "error_code" : "DCS.4961",
  "error_msg" : "Acl account request param invalid."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteAclAccountSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteAclAccountRequest request = new DeleteAclAccountRequest();
        request.withInstanceId("{instance_id}");
        request.withAccountId("{account_id}");
        try {
            DeleteAclAccountResponse response = client.deleteAclAccount(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```



```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteAclAccountRequest()
        request.instance_id = "{instance_id}"
        request.account_id = "{account_id}"
        response = client.delete_acl_account(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```
WithProjectId(projectId).
Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteAclAccountRequest{}
request.InstanceId = "{instance_id}"
request.AccountId = "{account_id}"
response, err := client.DeleteAclAccount(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ACL账号删除任务下发成功。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.10.7 ACL 账号修改备注

功能介绍

ACL账号修改备注。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/accounts/{account_id}

表 5-708 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
account_id	是	String	ACL账号ID。

请求参数

表 5-709 请求 Body 参数

参数	是否必选	参数类型	描述
description	否	String	备注信息。

响应参数

状态码：204

账号修改备注成功。

状态码：400

表 5-710 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-711 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-712 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-713 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-714 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/accounts/{account_id}
{
  "description" : "Test"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateAclAccountRemarkSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateAclAccountRemarkRequest request = new UpdateAclAccountRemarkRequest();
request.withInstanceId("{instance_id}");
request.withAccountId("{account_id}");
UpdateAclAccountRemarkRequestBody body = new UpdateAclAccountRemarkRequestBody();
body.withDescription("Test");
request.withBody(body);
try {
    UpdateAclAccountRemarkResponse response = client.updateAclAccountRemark(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAclAccountRemarkRequest()
        request.instance_id = "{instance_id}"
        request.account_id = "{account_id}"
```

```
request.body = UpdateAclAccountRemarkRequestBody(  
    description="Test"  
)  
response = client.update_acl_account_remark(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := dcs.NewDcsClient(  
        dcs.DcsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateAclAccountRemarkRequest{  
        request.InstanceId = "{instance_id}"  
        request.AccountId = "{account_id}"  
        descriptionUpdateAclAccountRemarkRequestBody := "Test"  
        request.Body = &model.UpdateAclAccountRemarkRequestBody{  
            Description: &descriptionUpdateAclAccountRemarkRequestBody,  
        }  
    }  
    response, err := client.UpdateAclAccountRemark(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	账号修改备注成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.11 网络安全

5.11.1 开启/关闭 SSL

功能介绍

开启/关闭SSL。该接口目前仅针对Redis 6.0/7.0基础版的单机、主备和Cluster集群实例。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/ssl

表 5-715 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-716 请求 Body 参数

参数	是否必选	参数类型	描述
enabled	是	Boolean	开启或关闭SSL。true：开启/ false：关闭

响应参数

状态码：200

表 5-717 响应 Body 参数

参数	参数类型	描述
job_id	String	DCS任务ID。
instance_id	String	实例ID。

状态码：400

表 5-718 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-719 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/ssl
{
  "enabled": true
}
```

响应示例

状态码：200

开启/关闭SSL成功。

```
{
  "job_id": "ff8080817fe01bb2017fe3cf68860481",
  "instance_id": "5560df16-cebf-4473-95c4-d1b573c16e79"
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4201",
  "error_msg": "Do not support SSL."
}
```

状态码：500

内部服务错误。

```
{
  "error_code": "DCS.5010",
  "error_msg": "Failed to operate SSL in database."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateSslSwitchSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSslSwitchRequest request = new UpdateSslSwitchRequest();
        request.withInstanceId("{instance_id}");
        UpdateSSLSwitchRequestBody body = new UpdateSSLSwitchRequestBody();
        body.withEnabled(true);
        request.withBody(body);
        try {
            UpdateSslSwitchResponse response = client.updateSslSwitch(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateSslSwitchRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateSSLSwitchRequestBody(
            enabled=True
        )
        response = client.update_ssl_switch(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSslSwitchRequest{}
```

```
request.InstanceId = "{instance_id}"
request.Body = &model.UpdateSslSwitchRequestBody{
    Enabled: true,
}
response, err := client.UpdateSslSwitch(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	开启/关闭SSL成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.11.2 查询实例 SSL 信息

功能介绍

查询实例SSL信息。该接口目前仅针对Redis 6.0/7.0基础版版本实例。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/ssl

表 5-720 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-721 响应 Body 参数

参数	参数类型	描述
enabled	Boolean	参数解释： 只有当实例开启了SSL时，返回该参数。 取值范围： true：实例已经开启了SSL。
ip	String	参数解释： SSL连接的IPv4地址。 取值范围： 不涉及。

参数	参数类型	描述
ipv6	String	参数解释： SSL连接的IPv6地址。 取值范围： 不涉及。
port	String	参数解释： SSL连接端口。 取值范围： 不涉及。
domain_name	String	参数解释： SSL连接域名。 取值范围： 不涉及。
ssl_expired_at	String	参数解释： SSL证书有效期（ UTC时间 ）。 取值范围： 不涉及。
ssl_validated	Boolean	参数解释： SSL证书是否有效。 取值范围： true：证书有效。 false：证书不可用。

状态码：204

参数解释：

无ssl信息。

状态码：400

表 5-722 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-723 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/ssl

响应示例

状态码：200

参数解释：

查询实例SSL信息成功。

```
{
  "enabled" : true,
  "ip" : "192.168.0.5",
  "ipv6" : "::1",
  "port" : "6379",
  "domain_name" : "redis-553a2516-b865-4159-b774-204dc0a36e37.dcs.example.com:6379",
  "ssl_expired_at" : "2022-06-15T02:21:18.669Z",
  "ssl_validated" : true
}
```

状态码：400

参数解释：

非法请求。

```
{
  "error_code" : "DCS.4201",
  "error_msg" : "Do not support SSL."
}
```

状态码：500

参数解释：

内部服务错误。

```
{
  "error_code" : "DCS.5010",
  "error_msg" : "Failed to operate SSL in database."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowInstanceSslDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
```

```

        .withCredential(auth)
        .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowInstanceSslDetailRequest request = new ShowInstanceSslDetailRequest();
    request.withInstanceId("{instance_id}");
    try {
        ShowInstanceSslDetailResponse response = client.showInstanceSslDetail(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceSslDetailRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_ssl_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"

```

```
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceSslDetailRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceSslDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 查询实例SSL信息成功。
204	参数解释： 无ssl信息。
400	参数解释： 非法请求。
500	参数解释： 内部服务错误。

错误码

请参见[错误码](#)。

5.11.3 下载实例 SSL 证书

功能介绍

下载实例SSL证书。该接口目前仅针对Redis 6.0/7.0基础版版本实例。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/ssl-certs/download

表 5-724 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码：200

表 5-725 响应 Body 参数

参数	参数类型	描述
file_name	String	SSL证书文件名。
link	String	SSL证书下载链接。
bucket_name	String	保存SSL证书的obs桶名。

状态码：400

表 5-726 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-727 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/ssl-certs/download
```

响应示例

状态码：200

下载实例SSL证书成功。

```
{
  "bucket_name" : "bucket_name",
  "file_name" : "file_name",
  "link" : "https://{bucket_name}-{obs_endpoint}:443/{ssl-XXX}/{file_name}?
  AWSSAccessKeyId=XXX&Expires=XXX&Signature=XXX"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4201",
  "error_msg" : "Do not support SSL."
}
```

状态码：500

内部服务错误。

```
{
  "error_code" : "DCS.5010",
  "error_msg" : "Failed to operate SSL in database."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DownloadSslCertSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
```

```
.withCredential(auth)
.withRegion(DcsRegion.valueOf("<YOUR REGION>"))
.build();
DownloadSslCertRequest request = new DownloadSslCertRequest();
request.withInstanceId("{instance_id}");
try {
    DownloadSslCertResponse response = client.downloadSslCert(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DownloadSslCertRequest()
        request.instance_id = "{instance_id}"
        response = client.download_ssl_cert(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
```



```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DownloadSslCertRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.DownloadSslCert(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下载实例SSL证书成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.11.4 设置 IP 白名单分组

功能介绍

为指定实例设置IP白名单分组，包含创建、停用、编辑、删除白名单四个功能。设置白名单分组信息会覆盖掉已有的白名单信息，因此在新增IP白名单分组时，需保留已有的白名单信息后再编辑新的白名单分组信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instance/{instance_id}/whitelist

表 5-728 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-729 请求 Body 参数

参数	是否必选	参数类型	描述
instance_id	否	String	参数解释： 实例ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
enable_whitelist	是	Boolean	参数解释： 是否启用白名单。 约束限制： 不涉及。 取值范围： true：启用白名单。 false：关闭白名单。 默认取值： 不涉及。
whitelist	是	Array of Whitelist objects	参数解释： IP白名单分组列表，设置白名单分组信息会覆盖掉已有的白名单信息，因此在新增IP白名单分组时，需保留已有的白名单信息后再编辑新的白名单分组信息。 约束限制： 不涉及。

表 5-730 Whitelist

参数	是否必选	参数类型	描述
group_name	是	String	参数解释： 白名单分组名称。 约束限制： 每个实例支持创建4个白名单分组，分组名称不能重复，必须大于四位，且首字段必须为字母，只能包含字母，数字，中划线，下划线。 取值范围： 不涉及。 默认取值： 不涉及。
ip_list	是	Array of strings	参数解释： 白名单分组下的IP列表。如果有多个，可以用逗号分隔。 约束限制： 单个实例下的IP不能重复，不支持IP部分为0.0.0.0的IP和CIDP。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：204

参数解释：

设置IP白名单成功。

状态码：400

表 5-731 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-732 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-733 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-734 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-735 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- 启用白名单，配置允许访问的IP地址。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist

{
  "enable_whitelist": true,
  "whitelist": [ {
    "group_name": "test001",
    "ip_list": [ "10.10.10.1", "10.10.10.2" ]
  } ]
}
```

- 启用白名单，配置允许访问的子网掩码。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

启用白名单，配置允许访问的IP地址。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateIpWhitelistSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateIpWhitelistRequest request = new UpdateIpWhitelistRequest();
        request.withInstanceId("{instance_id}");
        ModifyIpWhitelistBody body = new ModifyIpWhitelistBody();
        List<String> listWhitelistIpList = new ArrayList<>();
        listWhitelistIpList.add("10.10.10.1");
        listWhitelistIpList.add("10.10.10.2");
        List<Whitelist> listbodyWhitelist = new ArrayList<>();
        listbodyWhitelist.add(
            new Whitelist()
                .withGroupName("test001")
                .withIpList(listWhitelistIpList)
        );
        body.withWhitelist(listbodyWhitelist);
        body.withEnableWhitelist(true);
        request.withBody(body);
        try {
            UpdateIpWhitelistResponse response = client.updateIpWhitelist(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

启用白名单，配置允许访问的IP地址。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateIpWhitelistRequest()
        request.instance_id = "{instance_id}"
        listIpListWhitelist = [
            "10.10.10.1",
            "10.10.10.2"
        ]
        listWhitelistbody = [
            Whitelist(
                group_name="test001",
                ip_list=listIpListWhitelist
            )
        ]
        request.body = ModifyIpWhitelistBody(
            whitelist=listWhitelistbody,
            enable_whitelist=True
        )
        response = client.update_ip_whitelist(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

启用白名单，配置允许访问的IP地址。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```
auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateIpWhitelistRequest{}
request.InstanceId = "{instance_id}"
var listIpListWhitelist = []string{
    "10.10.10.1",
    "10.10.10.2",
}
var listWhitelistbody = []model.Whitelist{
    {
        GroupName: "test001",
        IpList: listIpListWhitelist,
    },
}
request.Body = &model.ModifyIpWhitelistBody{
    Whitelist: listWhitelistbody,
    EnableWhitelist: true,
}
response, err := client.UpdateIpWhitelist(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	参数解释： 设置IP白名单成功。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。

状态码	描述
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.11.5 查询指定实例的 IP 白名单

功能介绍

查询指定实例的IP白名单。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instance/{instance_id}/whitelist

表 5-736 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-737 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
enable_whitelist	Boolean	参数解释： 是否启用白名单。 取值范围： true：启用白名单。 false：未启用白名单。
whitelist	Array of WhitelistResp objects	参数解释： 实例当前IP白名单列表。
whitelist_ip_limit_num	String	参数解释： 租户实例添加IP白名单的最大IP数量限制。 取值范围： 不涉及。

表 5-738 WhitelistResp

参数	参数类型	描述
group_name	String	参数解释： 白名单分组名称。 取值范围： 不涉及。

参数	参数类型	描述
ip_list	Array of strings	参数解释： 白名单分组下的IP列表。 取值范围： 不涉及。

请求示例

GET https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist

响应示例

状态码：200

查询实例白名单成功。

```
{
  "instance_id": "5560df16-cebf-4473-95c4-d1b573c16e79",
  "enable_whitelist": true,
  "whitelist": [ {
    "group_name": "test001",
    "ip_list": [ "10.10.10.1", "10.10.10.2" ]
  } ],
  "whitelist_ip_limit_num": 20
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowIpWhitelistSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowIpWhitelistRequest request = new ShowIpWhitelistRequest();
request.withInstanceId("{instance_id}");
try {
    ShowIpWhitelistResponse response = client.showIpWhitelist(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowIpWhitelistRequest()
        request.instance_id = "{instance_id}"
        response = client.show_ip_whitelist(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowIpWhitelistRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowIpWhitelist(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例白名单成功。

错误码

请参见[错误码](#)。

5.11.6 设置 IP 白名单分组异步接口

功能介绍

为指定实例设置IP白名单分组，包含创建、停用、编辑、删除白名单四个功能。返回异步任务jobId，设置白名单分组信息会覆盖掉已有的白名单信息，因此在新增IP白名单分组时，需保留已有的白名单信息后再编辑新的白名单分组信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instance/{instance_id}/whitelist-async

表 5-739 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-740 请求 Body 参数

参数	是否必选	参数类型	描述
instance_id	否	String	参数解释： 实例ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
enable_whitelist	是	Boolean	参数解释： 是否启用白名单。 约束限制： 不涉及。 取值范围： true：启用白名单。 false：关闭白名单。 默认取值： 不涉及。
whitelist	是	Array of Whitelist objects	参数解释： IP白名单分组列表，设置白名单分组信息会覆盖掉已有的白名单信息，因此在新增IP白名单分组时，需保留已有的白名单信息后再编辑新的白名单分组信息。 约束限制： 不涉及。

表 5-741 Whitelist

参数	是否必选	参数类型	描述
group_name	是	String	参数解释： 白名单分组名称。 约束限制： 每个实例支持创建4个白名单分组，分组名称不能重复，必须大于四位，且首字段必须为字母，只能包含字母，数字，中划线，下划线。 取值范围： 不涉及。 默认取值： 不涉及。
ip_list	是	Array of strings	参数解释： 白名单分组下的IP列表。如果有多个，可以用逗号分隔。 约束限制： 单个实例下的IP不能重复，不支持IP部分为0.0.0.0的IP和CIDP。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-742 响应 Body 参数

参数	参数类型	描述
job_id	String	参数解释： 设置IP白名单任务ID。 取值范围： 不涉及。

状态码：400

表 5-743 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-744 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-745 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-746 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-747 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- 启用白名单，配置允许访问的IP地址。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist-async
{
  "enable_whitelist": true,
  "whitelist": [ {
    "group_name": "test001",
    "ip_list": [ "10.10.10.1", "10.10.10.2" ]
  } ]
}
```
- 启用白名单，配置允许访问的子网掩码。

```
PUT https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist-async
{
  "enable_whitelist": true,
  "whitelist": [ {
    "group_name": "test",
    "ip_list": [ "10.56.0.0/16" ]
  } ]
}
```

响应示例

状态码：200

参数解释：

设置IP白名单成功。

```
{
  "job_id": "job_id"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 启用白名单，配置允许访问的IP地址。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateIpWhitelistAsyncSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateIpWhitelistAsyncRequest request = new UpdateIpWhitelistAsyncRequest();
        request.withInstanceId("{instance_id}");
        ModifyIpWhitelistBody body = new ModifyIpWhitelistBody();
        List<String> listWhitelistIpList = new ArrayList<>();
        listWhitelistIpList.add("10.10.10.1");
        listWhitelistIpList.add("10.10.10.2");
        List<Whitelist> listbodyWhitelist = new ArrayList<>();
        listbodyWhitelist.add(
            new Whitelist()
                .withGroupName("test001")
                .withIpList(listWhitelistIpList)
        );
        body.withWhitelist(listbodyWhitelist);
        body.withEnableWhitelist(true);
        request.withBody(body);
        try {
            UpdateIpWhitelistAsyncResponse response = client.updateIpWhitelistAsync(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
        }
    }
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- 启用白名单，配置允许访问的子网掩码。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateIpWhitelistAsyncSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateIpWhitelistAsyncRequest request = new UpdateIpWhitelistAsyncRequest();
        request.withInstanceId("{instance_id}");
        ModifyIpWhitelistBody body = new ModifyIpWhitelistBody();
        List<String> listWhitelistIpList = new ArrayList<>();
        listWhitelistIpList.add("10.56.0.0/16");
        List<Whitelist> listbodyWhitelist = new ArrayList<>();
        listbodyWhitelist.add(
            new Whitelist()
                .withGroupName("test")
                .withIpList(listWhitelistIpList)
        );
        body.withWhitelist(listbodyWhitelist);
        body.withEnableWhitelist(true);
        request.withBody(body);
        try {
            UpdateIpWhitelistAsyncResponse response = client.updateIpWhitelistAsync(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- 启用白名单，配置允许访问的IP地址。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateIpWhitelistAsyncRequest()
        request.instance_id = "{instance_id}"
        listIpListWhitelist = [
            "10.10.10.1",
            "10.10.10.2"
        ]
        listWhitelistbody = [
            Whitelist(
                group_name="test001",
                ip_list=listIpListWhitelist
            )
        ]
        request.body = ModifyIpWhitelistBody(
            whitelist=listWhitelistbody,
            enable_whitelist=True
        )
        response = client.update_ip_whitelist_async(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 启用白名单，配置允许访问的子网掩码。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *
```



```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateIpWhitelistAsyncRequest()
        request.instance_id = "{instance_id}"
        listIpListWhitelist = [
            "10.56.0.0/16"
        ]
        listWhitelistbody = [
            Whitelist(
                group_name="test",
                ip_list=listIpListWhitelist
            )
        ]
        request.body = ModifyIpWhitelistBody(
            whitelist=listWhitelistbody,
            enable_whitelist=True
        )
        response = client.update_ip_whitelist_async(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 启用白名单，配置允许访问的IP地址。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```

    WithProjectId(projectId).
    Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

    request := &model.UpdateIpWhitelistAsyncRequest{}
    request.InstanceId = "{instance_id}"
    var listIpListWhitelist = []string{
        "10.10.10.1",
        "10.10.10.2",
    }
    var listWhitelistbody = []model.Whitelist{
        {
            GroupName: "test001",
            IpList: listIpListWhitelist,
        },
    }
    request.Body = &model.ModifyIpWhitelistBody{
        Whitelist: listWhitelistbody,
        EnableWhitelist: true,
    }
    response, err := client.UpdateIpWhitelistAsync(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- 启用白名单，配置允许访问的子网掩码。

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

    request := &model.UpdateIpWhitelistAsyncRequest{}
    request.InstanceId = "{instance_id}"
    var listIpListWhitelist = []string{

```

```
        "10.56.0.0/16",
    }
    var listWhitelistbody = []model.Whitelist{
        {
            GroupName: "test",
            IpList: listIpListWhitelist,
        },
    }
    request.Body = &model.ModifyIpWhitelistBody{
        Whitelist: listWhitelistbody,
        EnableWhitelist: true,
    }
    response, err := client.UpdateIpWhitelistAsync(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	参数解释： 设置IP白名单成功。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.11.7 添加 ip 白名单分组

功能介绍

为指定实例添加IP白名单分组。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instance/{instance_id}/whitelist-async

表 5-748 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-749 请求 Body 参数

参数	是否必选	参数类型	描述
whitelist	是	Array of Whitelist objects	参数解释： 需要添加的IP白名单分组列表,支持IP或CIDR。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-750 Whitelist

参数	是否必选	参数类型	描述
group_name	是	String	参数解释： 白名单分组名称。 约束限制： 每个实例支持创建4个白名单分组，分组名称不能重复，必须大于四位，且首字段必须为字母，只能包含字母，数字，中划线，下划线。 取值范围： 不涉及。 默认取值： 不涉及。
ip_list	是	Array of strings	参数解释： 白名单分组下的IP列表。如果有多个，可以用逗号分隔。 约束限制： 单个实例下的IP不能重复，不支持IP部分为0.0.0.0的IP和CIDP。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-751 响应 Body 参数

参数	参数类型	描述
job_id	String	参数解释： 添加IP白名单分组任务ID。 取值范围： 不涉及。

状态码：400

表 5-752 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-753 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-754 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-755 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-756 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instance/{instance_id}/whitelist-async
{
```



```
"whitelist": [ {  
  "group_name": "test001",  
  "ip_list": [ "10.10.10.1", "10.10.10.2" ]  
} ]  
}
```

响应示例

状态码：200

添加IP白名单分组返回体。

```
{  
  "job_id": "ff8080817fe01bb2017fe3cf68860481"  
}
```

状态码

状态码	描述
200	添加IP白名单分组返回体。
400	参数解释： 非法请求。
401	参数解释： 认证失败。
403	参数解释： 请求被拒绝。
404	参数解释： 路径错误。
500	参数解释： 内部错误。

错误码

请参见[错误码](#)。

5.12 后台任务管理

5.12.1 查询后台任务列表

功能介绍

查询后台任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/tasks

表 5-757 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

表 5-758 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量，表示从此偏移量开始查询，offset大于等于0。
limit	否	Integer	每页显示的条目数量。
start_time	否	String	查询开始时间，时间为UTC时间。格式：yyyyMMddHHmmss，如：20200609160000。
end_time	否	String	查询结束时间，时间为UTC时间。格式：yyyyMMddHHmmss，如：20200617155959。

请求参数

无

响应参数

状态码：200

表 5-759 响应 Body 参数

参数	参数类型	描述
task_count	String	任务个数。
tasks	Array of SingleBackgroundTask objects	任务详情数组。

表 5-760 SingleBackgroundTask

参数	参数类型	描述
id	String	后台任务ID。
name	String	后台任务名，目前支持以下取值： EXTEND：变更规格 BindEip：开启公网访问 UnBindEip：关闭公网访问 AddReplica：添加副本 DelReplica：删除副本 AddWhitelist：设置IP白名单 UpdatePort：修改端口 RemoveIpFromDns：域名摘除IP masterStandbySwapJob: 主备切换任务 modify：修改密码
details	DetailsBody object	详细信息。
user_name	String	用户名。
user_id	String	用户ID。
params	String	任务相关参数。
status	String	任务状态。
created_at	String	任务启动时间，格式为 2020-06-17T07:38:42.503Z。
updated_at	String	任务结束时间，格式为 2020-06-17T07:38:42.503Z。
enable_show	Boolean	是否有详细任务进展，可以展开查看。

表 5-761 DetailsBody

参数	参数类型	描述
old_capacity	String	变更前的容量，仅在变更规格时有值。
new_capacity	String	变更后的容量，仅在变更规格时有值。
enable_public_ip	Boolean	是否开启公网访问，仅在开启公网访问时有值。
public_ip_id	String	公网IP的ID，仅在开启公网访问时有值。
public_ip_address	String	公网IP地址，仅在开启公网访问时有值。
enable_ssl	Boolean	是否开启SSL，仅在开启SSL时有值。

参数	参数类型	描述
old_cache_mode	String	变更前的缓存类型，仅在变更规格时有值。
new_cache_mode	String	变更后的缓存类型，仅在变更规格时有值。
old_resource_spec_code	String	变更前的规格参数，仅在变更规格时有值。
new_resource_spec_code	String	变更后的规格参数，仅在变更规格时有值。
old_replica_num	Integer	变更前的副本数量，仅在变更规格时有值。
new_replica_num	Integer	变更后的副本数量，仅在变更规格时有值。
old_cache_type	String	变更前的缓存类型，仅在变更规格时有值。
new_cache_type	String	变更后的规格类型，仅在变更规格时有值。
replica_ip	String	副本IP。
replica_az	String	副本所在可用区。
group_name	String	组名。
old_port	Integer	旧端口。
new_port	Integer	新端口。
is_only_adjust_charging	Boolean	是否只是调整计费模式。
account_name	String	账号名称。
source_ip	String	源IP。
target_ip	String	目标IP。
node_name	String	节点信息。
rename_commands	Array of strings	重命名的指令。
updated_config_length	Integer	更新配置项的长度。

状态码：400

表 5-762 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-763 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-764 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-765 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-766 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-
cebf-4473-95c4-d1b573c16e79/tasks?
offset=0&limit=10&begin_time=20200609160000&end_time=20200617155959
```

响应示例

状态码：200

查询后台任务列表成功。

```
{
  "task_count": 1,
  "tasks": [ {
    "id": "8abf6e1e72c12e7c0172c15b508500aa",
    "name": "EXTEND",
    "user_name": "dcs_000_02",
    "user_id": "d53977d1adfb49c5b025ba7d33a13fd7",
    "details": {
      "old_cache_mode": "single",
      "new_cache_mode": "single",
      "old_capacity": "0.125",
      "new_capacity": "0.25",
      "old_resource_spec_code": "redis.single.xu1.tiny.125",
      "new_resource_spec_code": "redis.single.xu1.tiny.256",
      "old_replica_num": 1,
      "new_replica_num": 1,
      "old_cache_type": "single",
      "new_cache_type": "single",
      "replica_ip": "192.192.0.1",
      "replica_az": "az01",
      "group_name": "group1",
      "old_port": 6379,
      "new_port": 6379,
      "is_only_adjust_charging": false,
      "account_name": "account1",
      "source_ip": "192.192.0.0",
      "target_ip": "192.192.0.1",
      "node_name": "node1",
      "rename_commands": [ ],
      "updated_config_length": 0
    },
    "enable_show": false,
    "status": "SUCCESS",
    "created_at": "2020-06-17T08:18:06.597Z",
    "updated_at": "2020-06-17T08:18:10.106Z"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListBackgroundTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListBackgroundTaskRequest request = new ListBackgroundTaskRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListBackgroundTaskResponse response = client.listBackgroundTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBackgroundTaskRequest()
        request.instance_id = "{instance_id}"
        response = client.list_background_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBackgroundTaskRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListBackgroundTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```



```
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询后台任务列表成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.2 删除后台任务

功能介绍

删除后台任务。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}

表 5-767 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
task_id	是	String	后台任务ID。

请求参数

无

响应参数

状态码：200

表 5-768 响应 Body 参数

参数	参数类型	描述
message	String	返回消息。

状态码：204

删除后台任务成功。

状态码：400

表 5-769 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-770 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-771 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-772 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-773 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

删除后台任务。

```
DELETE https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/tasks/8abf6e1e72c12e7c0172c15b508500aa
```

响应示例

状态码：200

删除后台任务成功。

```
{  
  "message" : ""  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
public class DeleteBackgroundTaskSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        DeleteBackgroundTaskRequest request = new DeleteBackgroundTaskRequest();  
        request.withInstanceId("{instance_id}");  
        request.withTaskId("{task_id}");  
        try {  
            DeleteBackgroundTaskResponse response = client.deleteBackgroundTask(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBackgroundTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.delete_background_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.DeleteBackgroundTaskRequest{}
request.InstanceId = "{instance_id}"
request.TaskId = "{task_id}"
response, err := client.DeleteBackgroundTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除后台任务成功。
204	删除后台任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.3 查询后台任务详细信息

功能介绍

查询后台任务详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}/progress

表 5-774 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。
task_id	是	String	后台任务ID。

请求参数

无

响应参数

状态码：200

表 5-775 响应 Body 参数

参数	参数类型	描述
progress	Integer	总体进度，单位：%。
remain_time	Integer	剩余时间，单位：秒。
step_details	Array of StepDetail objects	任务详情列表。

表 5-776 StepDetail

参数	参数类型	描述
step_id	String	后台任务ID。
step_name	String	任务名。
step_status	String	任务状态，取值范围： - FINISH：完成 - FAILED：失败 - EXECUTING：执行中 - WAITING：等待中
begin_time	String	任务启动时间，格式为2020-06-17T07:38:42.503Z。
end_time	String	任务结束时间，格式为2020-06-17T07:38:42.503Z。

参数	参数类型	描述
error_code	String	错误码。
sub_step_details	Array of SubStepDetail objects	子任务详情列表。

表 5-777 SubStepDetail

参数	参数类型	描述
sub_step_id	String	任务ID。
sub_step_name	String	任务名。
sub_step_status	String	任务状态。
begin_time	String	任务启动时间，格式为 2020-06-17T07:38:42.503Z。
end_time	String	任务结束时间，格式为 2020-06-17T07:38:42.503Z。
detail	String	子任务的附加属性详情。
error_code	String	错误码。

状态码：400

表 5-778 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-779 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。

参数	参数类型	描述
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-780 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-781 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-782 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
GET https://{dcs_endpoint}/v2/a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71/instances/5560df16-cebf-4473-95c4-d1b573c16e79/tasks/8abf6e1e72c12e7c0172c15b508500aa/progress
```

响应示例

状态码：200

查询后台任务详情成功。

```
{
  "progress": 30,
  "remain_time": 100,
  "step_details": [ {
    "step_id": "9d1d1da8-650e-406f-81e4-b3c11515869f",
    "step_name": "前置检查",
    "step_status": "FINISH",
    "begin_time": "2022-09-01 11:52:24",
    "end_time": "2022-09-01 11:52:29",
    "error_code": "DCS.4064",
    "sub_step_details": [ {
      "sub_step_id": "4692b05b-fa55-4f7d-9be6-527bde25b424",
      "sub_step_name": "资源检查",
      "sub_step_status": "FINISH",
      "begin_time": "2022-09-01 11:52:24",
      "end_time": "2022-09-01 11:52:26",
      "detail": "",
      "error_code": "DCS.4065"
    }, {
      "sub_step_id": "4692b05b-fa55-4f7d-9be6-527bde25b425",
      "sub_step_name": "实例状态检查",
      "sub_step_status": "FINISH",
      "begin_time": "2022-09-01 11:52:26",
      "end_time": "2022-09-01 11:52:29",
      "detail": "",
      "error_code": "DCS.4064"
    } ]
  }, {
    "step_id": "9d1d1da8-650e-406f-3fad-c4a22515869f",
    "step_name": "数据迁移",
    "step_status": "EXECUTING",
    "begin_time": "2022-09-01 11:52:30",
    "end_time": "2022-09-01 11:54:30",
    "error_code": "DCS.4065",
    "sub_step_details": [ {
      "sub_step_id": "cb58a8ea-b8e2-4626-b3fa-57ab1a830724",
      "sub_step_name": "申请资源",
      "sub_step_status": "FINISH",
      "begin_time": "2022-09-01 11:52:30",
      "end_time": "2022-09-01 11:53:30",
      "detail": "{node_id:9996c3376318417281fb5e611f3fb1a0,slot_range:[0 5460],full_migration:迁移中,inc_migration:迁移中}",
      "error_code": "DCS.4065"
    }, {
      "sub_step_id": "f9df7fd2-b31f-4ec3-bbb1-3dc740326342",
      "sub_step_name": "申请资源",
      "sub_step_status": "FINISH",
      "begin_time": "2022-09-01 11:53:30",
      "end_time": "2022-09-01 11:54:30",
      "detail": "{node_id:5b70f440ccc94789b34a5c98c2ba05ea,slot_range:[5461 10921],full_migration:待开始,inc_migration:待开始}",
      "error_code": "DCS.4066"
    } ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowBackgroundTaskProgressSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBackgroundTaskProgressRequest request = new ShowBackgroundTaskProgressRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            ShowBackgroundTaskProgressResponse response = client.showBackgroundTaskProgress(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowBackgroundTaskProgressRequest()
    request.instance_id = "{instance_id}"
    request.task_id = "{task_id}"
    response = client.show_background_task_progress(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowBackgroundTaskProgressRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowBackgroundTaskProgress(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询后台任务详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.4 查询租户 Job 执行结果

功能介绍

查询租户Job执行结果。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/jobs/{job_id}

表 5-783 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
job_id	是	String	任务ID。

请求参数

无

响应参数

状态码：200

表 5-784 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。
job_type	String	Job类型，取值范围： - masterStandbySwapJob：主备倒换任务。 - modifyClientIpTransJob：修改源IP透传。 - modifyBandwidth：修改带宽任务。 - createDCSInstance：创建实例。 - closeDCSInstance：关闭实例。 - deleteDCSInstance：删除实例任务。 - backupDCSInstance：备份实例任务。 - restoreDCSInstance：恢复实例任务。 - extendDCSInstance：实例扩容任务。
status	String	Job状态，取值范围： - INIT：任务待执行。 - RUNNING：任务正在执行。 - SUCCESS：任务执行成功。 - FAIL：任务执行失败。
begin_time	String	Job开始时间戳，单位为ms，格式为:1684191545379。
end_time	String	Job开始时间戳，单位为ms，格式为:1684191548248。
fail_reason	String	失败原因。

状态码：400

表 5-785 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。

参数	参数类型	描述
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-786 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-787 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-788 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-789 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

GET https://{dcs_endpoint}/v2/{project_id}/jobs/{job_id}

响应示例

状态码：200

查询后台任务详情成功。

```
{
  "job_id": "ff8080818822bbf70188235afc24141a",
  "job_type": "masterStandbySwapJob",
  "status": "SUCCESS",
  "begin_time": "1684191545379",
  "end_time": "1684191548248",
  "fail_reason": null
}
```

状态码：400

非法请求。

```
{
  "error_code": "111400063",
  "error_msg": "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowJobInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```



```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowJobInfoRequest request = new ShowJobInfoRequest();
request.withJobId("{job_id}");
try {
    ShowJobInfoResponse response = client.showJobInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowJobInfoRequest()
        request.job_id = "{job_id}"
        response = client.show_job_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowJobInfoRequest{}
    request.JobId = "{job_id}"
    response, err := client.ShowJobInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询后台任务详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。

状态码	描述
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.5 查询任务中心任务列表

功能介绍

查询任务中心任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks

表 5-790 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

表 5-791 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量，表示从此偏移量开始查询，offset大于等于0。
limit	否	Integer	每页显示条数，最小值为1，最大值为1000，若不设置该参数，则为10。
start_time	否	String	查询开始时间，时间为UTC时间。格式：yyyyMMddHHmmss，如：20200609160000。

参数	是否必选	参数类型	描述
end_time	否	String	查询结束时间，时间为UTC时间。格式：yyyyMMddHHmmss，如：20230612155959。

请求参数

无

响应参数

状态码：200

表 5-792 响应 Body 参数

参数	参数类型	描述
task_count	String	任务个数。
tasks	Array of ListCenterTasksResp objects	中心任务详情数组。

表 5-793 ListCenterTasksResp

参数	参数类型	描述
id	String	后台任务ID。
name	String	后台任务名，目前支持以下取值： EXTEND：变更规格 BindEip：开启公网访问 UnBindEip：关闭公网访问 AddReplica：添加副本 DelReplica：删除副本 AddWhitelist：设置IP白名单 UpdatePort：修改端口 RemovelpFromDns：域名摘除IP masterStandbySwapJob: 主备切换任务 modify：修改密码
details	DetailsBody object	详细信息。
user_name	String	用户名。

参数	参数类型	描述
user_id	String	用户ID。
params	String	任务相关参数。
status	String	任务状态。
created_at	String	任务启动时间，格式为2020-06-17T07:38:42.503Z。
updated_at	String	任务结束时间，格式为2020-06-17T07:38:42.503Z。
error_code	String	错误代码。
enable_show	Boolean	是否有详细任务进展，可以展开查看。
job_id	String	任务ID。

表 5-794 DetailsBody

参数	参数类型	描述
old_capacity	String	变更前的容量，仅在变更规格时有值。
new_capacity	String	变更后的容量，仅在变更规格时有值。
enable_public_ip	Boolean	是否开启公网访问，仅在开启公网访问时有值。
public_ip_id	String	公网IP的ID，仅在开启公网访问时有值。
public_ip_address	String	公网IP地址，仅在开启公网访问时有值。
enable_ssl	Boolean	是否开启SSL，仅在开启SSL时有值。
old_cache_mode	String	变更前的缓存类型，仅在变更规格时有值。
new_cache_mode	String	变更后的缓存类型，仅在变更规格时有值。
old_resource_spec_code	String	变更前的规格参数，仅在变更规格时有值。
new_resource_spec_code	String	变更后的规格参数，仅在变更规格时有值。
old_replica_num	Integer	变更前的副本数量，仅在变更规格时有值。
new_replica_num	Integer	变更后的副本数量，仅在变更规格时有值。
old_cache_type	String	变更前的缓存类型，仅在变更规格时有值。

参数	参数类型	描述
new_cache_type	String	变更后的规格类型，仅在变更规格时有值。
replica_ip	String	副本IP。
replica_az	String	副本所在可用区。
group_name	String	组名。
old_port	Integer	旧端口。
new_port	Integer	新端口。
is_only_adjust_charging	Boolean	是否只是调整计费模式。
account_name	String	账号名称。
source_ip	String	源IP。
target_ip	String	目标IP。
node_name	String	节点信息。
rename_commands	Array of strings	重命名的指令。
updated_config_length	Integer	更新配置项的长度。

状态码：400

表 5-795 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-796 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。

参数	参数类型	描述
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/tasks
```

响应示例

状态码：200
查询任务中心任务成功。

```
{
  "task_count" : 2,
  "tasks" : [ {
    "id" : "8abf6e1e72c12e7c0172c15b508500aa",
    "name" : "ChangeInstanceSpec",
    "job_id" : "8abf6e1e87bce0740187bd32748b0530",
    "user_name" : "paas_dcs_a00421997_02",
    "user_id" : "d53977d1adfb49c5b025ba7d33a13fd7",
    "details" : {
      "old_cache_mode" : "single",
      "new_cache_mode" : "single",
      "old_capacity" : 0.125,
      "new_capacity" : 0.25
    },
    "status" : "SUCCESS",
    "created_at" : "2023-06-17T08:18:06.597Z",
    "updated_at" : "2023-06-17T08:18:10.106Z",
    "error_code" : "111400063",
    "enable_show" : true
  }, {
    "id" : "ff80808188bfde180188c36bb9ca4a30",
    "name" : "dcsExportJob",
    "job_id" : "ff80808188bfde180188c36bb9c44a2f",
    "user_name" : "paas_dcs_004_01",
    "user_id" : "bb259bd2b8b644a1b4a573a20318af51",
    "details" : null,
    "status" : "SUCCESS",
    "created_at" : "2023-06-17T08:18:06.597Z",
    "updated_at" : "2023-06-17T08:18:10.106Z",
    "error_code" : "111400063",
    "enable_show" : true
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListCenterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListCenterTaskRequest request = new ListCenterTaskRequest();
        try {
            ListCenterTaskResponse response = client.listCenterTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
```



```
.with_credentials(credentials) \
.with_region(DcsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ListCenterTaskRequest()
    response = client.list_center_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListCenterTaskRequest{}
    response, err := client.ListCenterTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询任务中心任务成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.6 删除任务中心任务

功能介绍

删除任务中心任务。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/tasks/{task_id}

表 5-797 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
task_id	是	String	后台任务ID。

请求参数

表 5-798 请求 Body 参数

参数	是否必选	参数类型	描述
message	否	String	删除结果

响应参数

状态码：200

删除后台任务成功。

状态码：400

表 5-799 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-800 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-801 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-802 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。

参数	参数类型	描述
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-803 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

DELETE https://{dcs_endpoint}/v2/{project_id}/tasks/{task_id}

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteCenterTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteCenterTaskRequest request = new DeleteCenterTaskRequest();
request.withTaskId("{task_id}");
try {
    DeleteCenterTaskResponse response = client.deleteCenterTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteCenterTaskRequest()
        request.task_id = "{task_id}"
        response = client.delete_center_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteCenterTaskRequest{}
    request.TaskId = "{task_id}"
    response, err := client.DeleteCenterTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除后台任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.7 提交前置检查任务

功能介绍

提交规格变更前置检查任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/resize/check-job

表 5-804 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-805 请求 Body 参数

参数	是否必选	参数类型	描述
new_capacity	否	Integer	新规格的容量，单位GB。
spec_code	否	String	新的规格编码。

响应参数

状态码：200

表 5-806 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。

状态码：400

表 5-807 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-808 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-809 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-810 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-811 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/resize/check-job
{
  "new_capacity": 1,
  "spec_code": "redis.single.xu1.large.1"
}
```

响应示例

状态码：200

提交检查任务成功。

```
{
  "job_id": "8abfa7aa88b4adbe0188b4d57a9206e6"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class StartInstanceResizeCheckJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
StartInstanceResizeCheckJobRequest request = new StartInstanceResizeCheckJobRequest();
request.withInstanceId("{instance_id}");
StartInstanceResizeCheckJobRequestBody body = new StartInstanceResizeCheckJobRequestBody();
body.withSpecCode("redis.single.xu1.large.1");
body.withNewCapacity(1);
request.withBody(body);
try {
    StartInstanceResizeCheckJobResponse response = client.startInstanceResizeCheckJob(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = StartInstanceResizeCheckJobRequest()
        request.instance_id = "{instance_id}"
        request.body = StartInstanceResizeCheckJobRequestBody(
            spec_code="redis.single.xu1.large.1",
            new_capacity=1
        )
        response = client.start_instance_resize_check_job(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StartInstanceResizeCheckJobRequest{
        request.InstanceId = "{instance_id}"
        specCodeStartInstanceResizeCheckJobRequestBody := "redis.single.xu1.large.1"
        newCapacityStartInstanceResizeCheckJobRequestBody := int32(1)
        request.Body = &model.StartInstanceResizeCheckJobRequestBody{
            SpecCode: &specCodeStartInstanceResizeCheckJobRequestBody,
            NewCapacity: &newCapacityStartInstanceResizeCheckJobRequestBody,
        }
    }
    response, err := client.StartInstanceResizeCheckJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	提交检查任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.8 查询实例列表导出任务详情

功能介绍

查询实例列表导出任务详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/export-job

表 5-812 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

表 5-813 Query 参数

参数	是否必选	参数类型	描述
job_id	否	String	实例列表导出任务ID。

请求参数

无

响应参数

状态码：200

表 5-814 响应 Body 参数

参数	参数类型	描述
status	String	任务状态。
file_name	String	文件名。
link	String	实例导出任务报告链接。
err_code	String	错误码。
err_msg	String	错误信息。

状态码：400

表 5-815 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-816 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-817 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-818 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-819 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/export-job?job_id={jobID}
```

响应示例

状态码：200

查询实例列表导出任务详情成功。

```
{
  "status" : "SUCCESS",
  "file_name" : "/tmp/Export/123.xls",
  "link" : "https://{obs_endpoint}/{project_id}/123",
  "err_code" : null,
```

```
"err_msg" : null
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ExportExcelJobSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ExportExcelJobRequest request = new ExportExcelJobRequest();
        try {
            ExportExcelJobResponse response = client.exportExcelJob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ExportExcelJobRequest()
        response = client.export_excel_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExportExcelJobRequest{}
    response, err := client.ExportExcelJob(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
```



```
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例列表导出任务详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.12.9 等待可维护时间窗取消规格变更

功能介绍

等待可维护时间窗取消规格变更。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/resize/cancel/{task_id}

表 5-820 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 任务ID。可通过DCS控制台进入后台任务界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-821 响应 Body 参数

参数	参数类型	描述
result	String	参数解释： 取消变更规格结果。 取值范围： success：取消变更规格成功。

状态码：400

表 5-822 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：401

表 5-823 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：403

表 5-824 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：404

表 5-825 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-826 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/resize/cancel/{task_id}
```

响应示例

状态码：200

取消规格变更请求成功。

```
{
  "result" : "success"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4239",
  "error_msg" : "The instance status is not extending."
}
```

状态码

状态码	描述
200	取消规格变更请求成功。

状态码	描述
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.13 实例诊断

5.13.1 创建实例诊断任务

功能介绍

诊断指定的缓存实例。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/diagnosis

表 5-827 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-828 请求 Body 参数

参数	是否必选	参数类型	描述
begin_time	是	String	诊断开始时间。UNIX时间戳，单位毫秒。
end_time	是	String	诊断结束时间。UNIX时间戳，单位毫秒。
node_ip_list	否	Array of strings	诊断节点IP列表。默认诊断所有节点。 非读写分离实例查询方法如下： - 方法一：参考查看实例信息。 - 方法二：调用查询指定实例接口查询。 读写分离实例查询方法：调用 查询分片信息 接口。

响应参数

状态码：200

表 5-829 响应 Body 参数

参数	参数类型	描述
report_id	String	报告ID。

状态码：400

表 5-830 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为null）。

状态码：500

表 5-831 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

创建实例诊断任务，设置诊断的起止时间和需要诊断的节点IP。

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/diagnosis
{
  "begin_time": 1618466522486,
  "end_time": 1618467122486,
  "node_ip_list": [ "192.168.0.111" ]
}
```

响应示例

状态码：200

下发诊断任务成功。

```
{
  "report_id": "716c9cca-8ce2-4a89-b3db-7cee209d6379"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建实例诊断任务，设置诊断的起止时间和需要诊断的节点IP。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateDiagnosisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateDiagnosisTaskRequest request = new CreateDiagnosisTaskRequest();
request.withInstanceId("{instance_id}");
CreateDiagnosisTaskBody body = new CreateDiagnosisTaskBody();
List<String> listbodyNodeIpList = new ArrayList<>();
listbodyNodeIpList.add("192.168.0.111");
body.withNodeIpList(listbodyNodeIpList);
body.withEndTime("1618467122486");
body.withBeginTime("1618466522486");
request.withBody(body);
try {
    CreateDiagnosisTaskResponse response = client.createDiagnosisTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建实例诊断任务，设置诊断的起止时间和需要诊断的节点IP。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = CreateDiagnosisTaskRequest()
    request.instance_id = "{instance_id}"
    listNodeIpListbody = [
        "192.168.0.111"
    ]
    request.body = CreateDiagnosisTaskBody(
        node_ip_list=listNodeIpListbody,
        end_time="1618467122486",
        begin_time="1618466522486"
    )
    response = client.create_diagnosis_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

创建实例诊断任务，设置诊断的起止时间和需要诊断的节点IP。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDiagnosisTaskRequest{}
    request.InstanceId = "{instance_id}"
    var listNodeIpListbody = []string{
        "192.168.0.111",
    }
    request.Body = &model.CreateDiagnosisTaskBody{
        NodeIpList: &listNodeIpListbody,
        EndTime: "1618467122486",
        BeginTime: "1618466522486",
    }
    response, err := client.CreateDiagnosisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下发诊断任务成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.13.2 查询实例诊断任务列表

功能介绍

查询指定缓存实例诊断任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/diagnosis

表 5-832 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

表 5-833 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示条数，最小值为1，最大值为1000，若不设置该参数，则为10。
offset	否	Integer	偏移量，表示从此偏移量开始查询，offset大于等于0。

请求参数

无

响应参数

状态码：200

表 5-834 响应 Body 参数

参数	参数类型	描述
diagnosis_report_list	Array of DiagnosisReportInfo objects	诊断报告列表。
total_num	Integer	诊断报告总数。

表 5-835 DiagnosisReportInfo

参数	参数类型	描述
report_id	String	诊断报告ID。
status	String	诊断任务状态，diagnosing：诊断中，finished：完成诊断。
begin_time	String	诊断时间段的开始时间。格式为：2017-03-31T12:24:46.297Z。
end_time	String	诊断时间段的结束时间。格式为：2017-03-31T12:24:46.297Z。
created_at	String	诊断报告创建时间。
node_num	Integer	参与诊断的节点个数。
abnormal_item_sum	Integer	诊断结果为异常的诊断项总数。
failed_item_sum	Integer	诊断失败的诊断项总数。

状态码：400

表 5-836 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-837 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

无

响应示例

状态码：200

查询实例诊断报告列表成功。

```
{
  "diagnosis_report_list": [ {
    "report_id": "fb98e37a-3cd9-4c7a-96b1-d878266019e2",
    "status": "finished",
    "begin_time": "2021-04-15T06:02:00.000Z",
    "end_time": "2021-04-15T06:12:00.000Z",
    "created_at": "2021-04-15T06:12:06.641Z",
    "abnormal_item_sum": 2,
    "failed_item_sum": 0,
    "node_num": 2
  } ],
  "total_num": 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListDiagnosisTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDiagnosisTasksRequest request = new ListDiagnosisTasksRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListDiagnosisTasksResponse response = client.listDiagnosisTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListDiagnosisTasksRequest()
    request.instance_id = "{instance_id}"
    response = client.list_diagnosis_tasks(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDiagnosisTasksRequest{
        request.InstanceId = "{instance_id}"
    }
    response, err := client.ListDiagnosisTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询实例诊断报告列表成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.13.3 删除诊断记录

功能介绍

删除诊断记录。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/diagnosis

表 5-838 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-839 请求 Body 参数

参数	是否必选	参数类型	描述
report_id_list	否	Array of strings	实例诊断报告ID

响应参数

状态码：200

表 5-840 响应 Body 参数

参数	参数类型	描述
results	Array of Item objects	删除实例诊断结果

表 5-841 Item

参数	参数类型	描述
result	String	诊断结果
id	String	诊断报告ID

状态码：400

表 5-842 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-843 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/diagnosis
{
  "report_id_list" : [ "5c4a78a3-b502-4b17-9a5b-6fb955c307bd" ]
}
```

响应示例

状态码：200
删除诊断记录成功。

```
{
  "results" : [ {
    "result" : "successful",
    "id" : "5c4a78a3-b502-4b17-9a5b-6fb955c307bd"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteDiagnosisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteDiagnosisTaskRequest request = new DeleteDiagnosisTaskRequest();
request.withInstanceId("{instance_id}");
DeleteDiagnosisReportRequest body = new DeleteDiagnosisReportRequest();
List<String> listbodyReportIdList = new ArrayList<>();
listbodyReportIdList.add("5c4a78a3-b502-4b17-9a5b-6fb955c307bd");
body.withReportIdList(listbodyReportIdList);
request.withBody(body);
try {
    DeleteDiagnosisTaskResponse response = client.deleteDiagnosisTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDiagnosisTaskRequest()
        request.instance_id = "{instance_id}"
        listReportIdListbody = [
            "5c4a78a3-b502-4b17-9a5b-6fb955c307bd"
        ]
        request.body = DeleteDiagnosisReportRequest(
```

```
        report_id_list=listReportIdListbody
    )
    response = client.delete_diagnosis_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDiagnosisTaskRequest{}
    request.InstanceId = "{instance_id}"
    var listReportIdListbody = []string{
        "5c4a78a3-b502-4b17-9a5b-6fb955c307bd",
    }
    request.Body = &model.DeleteDiagnosisReportRequest{
        ReportIdList: &listReportIdListbody,
    }
    response, err := client.DeleteDiagnosisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除诊断记录成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.13.4 查询指定诊断报告

功能介绍

通过报告ID查询诊断报告的详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/diagnosis/{report_id}

表 5-844 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
report_id	是	String	诊断报告ID。

请求参数

无

响应参数

状态码：200

表 5-845 响应 Body 参数

参数	参数类型	描述
abnormal_item_sum	Integer	诊断结果为异常的诊断项总数。
failed_item_sum	Integer	诊断失败的诊断项总数。
diagnosis_node_report_list	Array of DiagnosisNodeReport objects	节点诊断报告列表。

表 5-846 DiagnosisNodeReport

参数	参数类型	描述
node_ip	String	节点IP。例如：192.168.0.234:6379。
az_code	String	节点所在可用区Code。
group_name	String	节点所在分片的名称。
abnormal_sum	Integer	诊断结果为异常的诊断项总数。
failed_sum	Integer	诊断失败的诊断项总数。
role	String	节点角色，master：主节点，slave：备节点。
diagnosis_dimension_list	Array of DiagnosisDimension objects	诊断维度列表。
command_time_taken_list	CommandTimeTakenList object	命令耗时统计列表。

表 5-847 DiagnosisDimension

参数	参数类型	描述
name	String	诊断维度名称。
abnormal_num	Integer	诊断结果为异常的诊断项总数。
failed_num	Integer	诊断失败的诊断项总数。
diagnosis_item_list	Array of DiagnosisItem objects	诊断项列表。

表 5-848 DiagnosisItem

参数	参数类型	描述
name	String	诊断项名称。
cause_ids	Array of ConclusionItem objects	原因ID列表，ID说明请见 实例诊断ID说明 。
impact_ids	Array of ConclusionItem objects	影响ID列表，ID说明请见 实例诊断ID说明 。
advice_ids	Array of ConclusionItem objects	建议ID列表，ID说明请见 实例诊断ID说明 。
result	String	诊断结果。

表 5-849 ConclusionItem

参数	参数类型	描述
id	Integer	结论ID。
params	Map<String,String>	结论参数。

表 5-850 CommandTimeTakenList

参数	参数类型	描述
total_num	Integer	执行命令的总次数。
total_usec_sum	Double	执行命令的总耗时。
result	String	命令耗时统计结果。
command_list	Array of CommandTimeTaken objects	命令耗时统计。

表 5-851 CommandTimeTaken

参数	参数类型	描述
calls_sum	Integer	调用次数。
usec_sum	Double	耗时总数。

参数	参数类型	描述
command_name	String	命令名称。
per_usec	String	耗时占比。
average_usec	Double	每次调用平均耗时。

状态码：400

表 5-852 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

状态码：500

表 5-853 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息（暂未使用，赋值为 null）。

请求示例

无

响应示例

状态码：200

查询指定报告成功。

```
{
  "abnormal_item_sum" : 1,
  "failed_item_sum" : 0,
  "diagnosis_node_report_list" : [ {
    "group_name" : "group-0",
    "az_code" : "region01",
    "node_ip" : "192.168.0.170:6379",
    "abnormal_sum" : 1,
```

```
"failed_sum" : 0,
"role" : "master",
"command_time_taken_list" : {
  "command_list" : [ {
    "calls_sum" : 329,
    "usec_sum" : 20.732,
    "command_name" : "info",
    "per_usec" : "68.61%",
    "average_usec" : 0.063
  }, {
    "calls_sum" : 1788,
    "usec_sum" : 1.787,
    "command_name" : "ping",
    "per_usec" : "5.91%",
    "average_usec" : 0.001
  }, {
    "calls_sum" : 2,
    "usec_sum" : 0.025,
    "command_name" : "config",
    "per_usec" : "0.08%",
    "average_usec" : 0.013
  }, {
    "calls_sum" : 60,
    "usec_sum" : 0.186,
    "command_name" : "slowlog",
    "per_usec" : "0.62%",
    "average_usec" : 0.003
  }, {
    "calls_sum" : 1764,
    "usec_sum" : 7.485,
    "command_name" : "publish",
    "per_usec" : "24.77%",
    "average_usec" : 0.004
  } ],
  "result" : "succeed",
  "error_code" : null,
  "total_num" : 5,
  "total_usec_sum" : 30.215
},
"diagnosis_dimension_list" : [ {
  "name" : "load",
  "abnormal_num" : 0,
  "failed_num" : 0,
  "diagnosis_item_list" : [ {
    "name" : "cpu_usage",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  }, {
    "name" : "time_consuming_commands",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  } ]
}, {
  "name" : "storage",
  "abnormal_num" : 1,
  "failed_num" : 0,
  "diagnosis_item_list" : [ {
    "name" : "inner_memory_fragmentation",
    "result" : "normal",
    "cause_ids" : [ {
      "id" : 7,
      "params" : null
    } ],
  } ],
}
```



```

    "impact_ids" : [ {
      "id" : 3,
      "params" : null
    } ],
    "advice_ids" : [ {
      "id" : 4,
      "params" : null
    } ],
    "error_code" : null
  }, {
    "name" : "persistence",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  }, {
    "name" : "centralized_expiration",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  }, {
    "name" : "memory_usage",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  }, {
    "name" : "hit_ratio",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  } ]
}, {
  "name" : "network",
  "abnormal_num" : 0,
  "failed_num" : 0,
  "diagnosis_item_list" : [ {
    "name" : "connection_num",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  }, {
    "name" : "rx_controlled",
    "result" : "normal",
    "cause_ids" : null,
    "impact_ids" : null,
    "advice_ids" : null,
    "error_code" : null
  } ]
} ]
}

```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowDiagnosisTaskDetailsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDiagnosisTaskDetailsRequest request = new ShowDiagnosisTaskDetailsRequest();
        request.withReportId("{report_id}");
        try {
            ShowDiagnosisTaskDetailsResponse response = client.showDiagnosisTaskDetails(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowDiagnosisTaskDetailsRequest()
    request.report_id = "{report_id}"
    response = client.show_diagnosis_task_details(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDiagnosisTaskDetailsRequest{}
    request.ReportId = "{report_id}"
    response, err := client.ShowDiagnosisTaskDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询指定报告成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.14 模板管理

5.14.1 查询参数模板列表

功能介绍

查询租户的参数模板列表，支持按照条件查询。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/config-templates

表 5-854 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-855 Query 参数

参数	是否必选	参数类型	描述
name	否	String	参数解释： 参数模板名称，支持模糊查找。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
type	是	String	参数解释： 模板类型。 约束限制： 不涉及。 取值范围： - sys：系统默认模板。 - user：用户自定义模板。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
engine	否	String	参数解释： 缓存引擎。 约束限制： 不涉及。 取值范围： • Redis • Memcached 默认取值： 不涉及。
engine_version	否	String	参数解释： 缓存版本。 约束限制： 不涉及。 取值范围： 当缓存引擎为Redis时，取值为4.0/5.0/6.0/7.0。 当缓存引擎为Memcached时，该字段为可选，取值为空。 默认取值： 不涉及。
cache_mode	否	String	参数解释： 缓存实例类型。 约束限制： 不涉及。 取值范围： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例 默认取值： 不涉及。

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示从此偏移量开始查询。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示条数。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10

请求参数

无

响应参数

状态码：200

表 5-856 响应 Body 参数

参数	参数类型	描述
template_num	Integer	参数解释： 模板个数。 取值范围： 不涉及。
templates	Array of ConfigTemplates ListInfo objects	参数解释： 模板的详情数组。 取值范围： 不涉及。

表 5-857 ConfigTemplatesListInfo

参数	参数类型	描述
template_id	String	参数解释： 模板ID。 取值范围： 不涉及。
cache_mode	String	参数解释： 缓存实例类型。 取值范围： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例
description	String	参数解释： 模板的描述信息。 取值范围： 不涉及。
engine	String	参数解释： 缓存引擎。 取值范围： • Redis • Memcached
engine_version	String	参数解释： 缓存版本。 取值范围： 当缓存引擎为Redis时，取值为4.0/5.0/6.0/7.0。 当缓存引擎为Memcached时，该字段为可选，取值为空。
name	String	参数解释： 模板名称。 取值范围： 不涉及。

参数	参数类型	描述
product_type	String	参数解释： 产品类型。 取值范围： - generic：基础版 - enterprise：企业版
storage_type	String	参数解释： 存储类型。 取值范围： - DRAM - SSD
type	String	参数解释： 模板类型。 取值范围： - sys：系统默认模板。 - user：用户自定义模板。
created_at	String	参数解释： 模板创建的时间，仅在自定义参数模板中有意义，格式：2023-05-10T11:09:35.802Z。 取值范围： 不涉及。

状态码：400

表 5-858 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：401

表 5-859 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：403

表 5-860 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：404

表 5-861 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：500

表 5-862 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/config-templates?
type=sys&engine=Redis&engine_version=5.0&cache_mode=ha&offset=0&limit=10
```

响应示例

状态码：200
查询参数模板列表成功。

```
{
  "template_num": 1,
  "templates": [ {
    "template_id": "6",
    "cache_mode": "single",
    "description": null,
    "engine": "Redis",
    "engine_version": "4.0",
    "name": "Default-Redis-4.0-single-generic-DRAM",
    "product_type": "generic",
    "storage_type": "DRAM",
    "type": "sys"
  } ]
}
```

状态码：400
非法请求。

```
{
  "error_code": "111400063",
  "error_msg": "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class ListConfigTemplatesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConfigTemplatesRequest request = new ListConfigTemplatesRequest();
        try {
            ListConfigTemplatesResponse response = client.listConfigTemplates(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListConfigTemplatesRequest()
        response = client.list_config_templates(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListConfigTemplatesRequest{}
    response, err := client.ListConfigTemplates(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询参数模板列表成功。
400	非法请求。
401	认证信息非法。

状态码	描述
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.14.2 创建自定义模板

功能介绍

创建自定义模板。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/config-templates

表 5-863 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

请求参数

表 5-864 请求 Body 参数

参数	是否必选	参数类型	描述
template_id	是	String	来源系统模板ID。
name	是	String	模板名称。
type	是	String	模板类型，取值范围： - sys：系统默认模板 - user：用户自定义模板
engine	否	String	缓存引擎：Redis和Memcached。

参数	是否必选	参数类型	描述
cache_mode	否	String	缓存实例类型。取值范围如下： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例
description	否	String	模板的描述信息。
engine_version	否	String	缓存版本。 当缓存引擎为Redis时，取值为4.0/5.0/6.0/7.0。 当缓存引擎为Memcached时，该字段为可选，取值为空。
params	是	Map<String,String>	参数配置信息。

响应参数

状态码：200

表 5-865 响应 Body 参数

参数	参数类型	描述
id	String	模板ID。

状态码：400

表 5-866 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：401

表 5-867 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：403

表 5-868 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：404

表 5-869 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：500

表 5-870 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

请求示例

创建一个Redis5.0单机实例的自定义模板。

```
POST https://{dcs_endpoint}/v2/{project_id}/config-templates
{
  "name" : "Test",
  "cache_mode" : "single",
  "description" : "Test",
  "engine" : "Redis",
  "engine_version" : "5.0",
  "params" : {
    "latency-monitor-threshold" : 15
  },
  "template_id" : "11",
  "type" : "sys"
}
```

响应示例

状态码：200

创建自定义模板成功。

```
{
  "id" : "efb1ba06-d3cd-4a77-9173-16f70f2d1343"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个Redis5.0单机实例的自定义模板。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateCustomTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCustomTemplateRequest request = new CreateCustomTemplateRequest();
        CreateCustomTemplateBody body = new CreateCustomTemplateBody();
        Map<String, String> listbodyParams = new HashMap<>();
        listbodyParams.put("latency-monitor-threshold", "15");
        body.withParams(listbodyParams);
        body.withEngineVersion("5.0");
        body.withDescription("Test");
        body.withCacheMode("single");
        body.withEngine("Redis");
        body.withType("sys");
        body.withName("Test");
        body.withTemplateId("11");
        request.withBody(body);
        try {
            CreateCustomTemplateResponse response = client.createCustomTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建一个Redis5.0单机实例的自定义模板。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateCustomTemplateRequest()
        listParamsbody = {
            "latency-monitor-threshold": "15"
        }
        request.body = CreateCustomTemplateBody(
            params=listParamsbody,
            engine_version="5.0",
            description="Test",
            cache_mode="single",
            engine="Redis",
            type="sys",
            name="Test",
            template_id="11"
        )
        response = client.create_custom_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建一个Redis5.0单机实例的自定义模板。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
```

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateCustomTemplateRequest{
var listParamsbody = map[string]string{
    "latency-monitor-threshold": "15",
}
engineVersionCreateCustomTemplateBody:= "5.0"
descriptionCreateCustomTemplateBody:= "Test"
cacheModeCreateCustomTemplateBody:= "single"
engineCreateCustomTemplateBody:= "Redis"
request.Body = &model.CreateCustomTemplateBody{
    Params: listParamsbody,
    EngineVersion: &engineVersionCreateCustomTemplateBody,
    Description: &descriptionCreateCustomTemplateBody,
    CacheMode: &cacheModeCreateCustomTemplateBody,
    Engine: &engineCreateCustomTemplateBody,
    Type: "sys",
    Name: "Test",
    TemplateId: "11",
}
response, err := client.CreateCustomTemplate(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建自定义模板成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.14.3 查询参数模板详情

功能介绍

查询参数模板详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/config-templates/{template_id}

表 5-871 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
template_id	是	String	模板ID。

表 5-872 Query 参数

参数	是否必选	参数类型	描述
type	是	String	模板类型，取值范围： - sys：系统默认模板 - user：用户自定义模板

请求参数

无

响应参数

状态码：200

表 5-873 响应 Body 参数

参数	参数类型	描述
template_id	String	参数解释： 模板ID。 取值范围： 不涉及。
cache_mode	String	参数解释： 缓存实例类型。 取值范围： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例
description	String	参数解释： 模板的描述信息。 取值范围： 不涉及。
engine	String	参数解释： 缓存引擎。 取值范围： • Redis • Memcached
engine_version	String	参数解释： 缓存版本。 取值范围： 当缓存引擎为Redis时，取值为4.0/5.0/6.0/7.0。 当缓存引擎为Memcached时，该字段为可选，取值为空。
name	String	参数解释： 模板名称。 取值范围： 不涉及。

参数	参数类型	描述
product_type	String	参数解释： 产品类型。 取值范围： - generic：基础版 - enterprise：企业版
storage_type	String	参数解释： 存储类型。 取值范围： - DRAM - SSD
type	String	参数解释： 模板类型。 取值范围： - sys：系统默认模板。 - user：用户自定义模板。
created_at	String	参数解释： 模板创建的时间，仅在自定义参数模板中有意义，格式：2023-05-10T11:09:35.802Z。 取值范围： 不涉及。

状态码：400

表 5-874 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：401

表 5-875 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：403

表 5-876 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：404

表 5-877 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：500

表 5-878 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

请求示例

GET https://{dcs_endpoint}/v2/{project_id}/config-templates/{template_id}?type=user

响应示例

状态码：200

查询参数模板列表成功。

```
{
  "template_id" : "bfa904a1-27a4-4387-9f0e-44a524c8a244",
  "cache_mode" : "ha_rw_split",
  "created_at" : "2023-05-10T11:46:21.290Z",
  "description" : null,
  "engine" : "Redis",
  "engine_version" : "4.0",
  "name" : "test",
  "product_type" : "generic",
  "storage_type" : "DRAM",
  "type" : "user"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowConfigTemplateSolution {

    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowConfigTemplateRequest request = new ShowConfigTemplateRequest();
request.withTemplateId("{template_id}");
try {
    ShowConfigTemplateResponse response = client.showConfigTemplate(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConfigTemplateRequest()
        request.template_id = "{template_id}"
        response = client.show_config_template(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConfigTemplateRequest{}
    request.TemplateId = "{template_id}"
    response, err := client.ShowConfigTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询参数模板列表成功。
400	非法请求。
401	认证信息非法。

状态码	描述
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.14.4 修改自定义模板

功能介绍

修改自定义模板。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/config-templates/{template_id}

表 5-879 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
template_id	是	String	模板ID。

请求参数

表 5-880 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	模板名称。
description	否	String	模板的描述信息。
params	是	Map<String,String>	参数配置信息。

响应参数

状态码：200

表 5-881 响应 Body 参数

参数	参数类型	描述
result	String	修改自定义模板的结果，success表示成功，如果失败则会返回对应的报错信息。

状态码：400

表 5-882 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：401

表 5-883 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：403

表 5-884 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：404

表 5-885 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：500

表 5-886 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/config-templates/{template_id}

{
  "name" : "Test",
  "description" : "Test",
  "params" : {
    "latency-monitor-threshold" : 15
  }
}
```

响应示例

状态码：200

修改自定义模板成功。

```
{  
  "result" : "success"  
}
```

状态码：400

非法请求。

```
{  
  "error_code" : "111400063",  
  "error_msg" : "Invalid {0} parameter in the request."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
import java.util.Map;  
import java.util.HashMap;  
  
public class UpdateConfigTemplateSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        UpdateConfigTemplateRequest request = new UpdateConfigTemplateRequest();  
        request.withTemplateId("{template_id}");  
        UpdateCustomTemplateBody body = new UpdateCustomTemplateBody();  
        Map<String, String> listbodyParams = new HashMap<>();  
        listbodyParams.put("latency-monitor-threshold", "15");  
        body.withParams(listbodyParams);  
        body.withDescription("Test");  
        body.withName("Test");  
        request.withBody(body);  
        try {  
            UpdateConfigTemplateResponse response = client.updateConfigTemplate(request);  
            System.out.println(response.toString());  
        }  
    }  
}
```

```

    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}

```

Python

```

# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateConfigTemplateRequest()
        request.template_id = "{template_id}"
        listParamsbody = {
            "latency-monitor-threshold": "15"
        }
        request.body = UpdateCustomTemplateBody(
            params=listParamsbody,
            description="Test",
            name="Test"
        )
        response = client.update_config_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"

```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateConfigTemplateRequest{}
    request.TemplateId = "{template_id}"
    var listParamsbody = map[string]string{
        "latency-monitor-threshold": "15",
    }
    descriptionUpdateCustomTemplateBody:= "Test"
    nameUpdateCustomTemplateBody:= "Test"
    request.Body = &model.UpdateCustomTemplateBody{
        Params: listParamsbody,
        Description: &descriptionUpdateCustomTemplateBody,
        Name: &nameUpdateCustomTemplateBody,
    }
    response, err := client.UpdateConfigTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	修改自定义模板成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.14.5 删除自定义模板

功能介绍

删除自定义模板。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/config-templates/{template_id}

表 5-887 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
template_id	是	String	模板ID。

请求参数

无

响应参数

状态码：200

表 5-888 响应 Body 参数

参数	参数类型	描述
result	String	删除自定义模板的结果，success表示成功，如果失败则会返回对应的报错信息。

状态码：400

表 5-889 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：401

表 5-890 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：403

表 5-891 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：404

表 5-892 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

状态码：500

表 5-893 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息(暂未使用，赋值为null)。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/config-templates/{template_id}
```

响应示例

状态码：200

删除自定义模板成功。

```
{
  "result" : "success"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
```

```
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteConfigTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteConfigTemplateRequest request = new DeleteConfigTemplateRequest();
        request.withTemplateId("{template_id}");
        try {
            DeleteConfigTemplateResponse response = client.deleteConfigTemplate(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = DeleteConfigTemplateRequest()
    request.template_id = "{template_id}"
    response = client.delete_config_template(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteConfigTemplateRequest{}
    request.TemplateId = "{template_id}"
    response, err := client.DeleteConfigTemplate(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除自定义模板成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15 其他接口

5.15.1 查询产品规格

功能介绍

在创建缓存实例时，需要配置订购的产品规格编码（spec_code），可通过该接口查询产品规格，查询条件不选时默认查询全部。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/flavors

表 5-894 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-895 Query 参数

参数	是否必选	参数类型	描述
instance_id	否	String	参数解释： 缓存实例ID，可通过DCS控制台进入实例详情界面查看。根据该字段可查询出实例可变更的规格。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
spec_code	否	String	参数解释： 产品规格编码。获取方法如下： - 方法一：登录分布式缓存的控制台，在创建缓存实例页面，查找对应的实例规格的规格编码。 - 方法二：调用查询产品规格接口查询。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
cache_mode	否	String	参数解释： 缓存实例类型。 约束限制： 不涉及。 取值范围： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例 默认取值： 不涉及。
engine	否	String	参数解释： 缓存引擎类型。 约束限制： 不涉及。 取值范围： • Redis • Memcached 默认取值： 不涉及。

参数	是否必选	参数类型	描述
engine_version	否	String	参数解释： 缓存版本。 约束限制： 不涉及。 取值范围： Redis缓存引擎的取值范围如下： • 3.0 • 4.0 • 5.0 • 6.0 • 7.0 当缓存引擎为Memcached时，取值为空。 默认取值： 不涉及。
cpu_type	否	String	参数解释： CPU架构类型。 约束限制： 不涉及。 取值范围： • x86_64：X86架构。 • aarch64：ARM架构。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
capacity	否	String	参数解释： 缓存容量（GB）。 约束限制： 不涉及。 取值范围： - Redis4.0/5.0/6.0/7.0：单机和主备类型实例取值：0.125、0.25、0.5、1、2、4、8、16、24、32、48、64。集群实例规格支持4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024。读写分离实例规格支持1、2、4、8、16、32、64。 - Memcached：单机和主备类型实例取值：2、4、8、16、32、64。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-896 响应 Body 参数

参数	参数类型	描述
flavors	Array of FlavorsItems objects	参数解释： 产品规格详情。 取值范围： 不涉及。

表 5-897 FlavorsItems

参数	参数类型	描述
spec_code	String	参数解释： 产品规格编码。 取值范围： 不涉及。
cloud_service_type_code	String	参数解释： 云服务类型编码。 取值范围： 不涉及。
cloud_resource_type_code	String	参数解释： 云资源类型编码。 取值范围： 不涉及。
cache_mode	String	参数解释： 缓存实例类型。 取值范围： • single：表示单机实例 • ha：表示主备实例 • cluster：表示cluster集群实例 • proxy：表示Proxy集群实例 • ha_rw_split：表示读写分离实例
engine	String	参数解释： 缓存引擎类型。 取值范围： • Redis • Memcached
engine_version	String	参数解释： 缓存版本。 取值范围： Redis缓存引擎的取值范围如下： • 3.0 • 4.0 • 5.0 • 6.0 • 7.0 当缓存引擎为Memcached时，取值为空。

参数	参数类型	描述
product_type	String	参数解释： 缓存实例的产品类型。 取值范围： 取值当前仅支持generic：标准类型。
cpu_type	String	参数解释： CPU架构类型。 取值范围： - x86_64：X86架构。 - aarch64：ARM架构。
storage_type	String	参数解释： 存储类型。 取值范围： 取值当前仅支持DRAM：内存存储。
capacity	Array of strings	参数解释： 缓存容量（GB）。 取值范围： 0.125、0.25、0.5、1、2、4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024、2048等。
billing_mode	Array of strings	参数解释： 计费模式。 取值范围： - Hourly：按需计费 - Monthly：包月计费 - Yearly：包周期计费
tenant_ip_count	Integer	参数解释： 租户侧IP数量。 取值范围： 不涉及。
pricing_type	String	参数解释： 定价类型。 取值范围： - tier：阶梯定价，一个规格对应多个容量。 - normal：规格和容量一一对应。

参数	参数类型	描述
is_dec	Boolean	参数解释： 是否支持专属云。 取值范围： • true • false
attrs	Array of AttrsObject objects	参数解释： 规格的其他信息。 取值范围： 不涉及。
flavors_available_zones	Array of FlavorAzObject objects	参数解释： 有资源的可用区。 取值范围： 不涉及。
replica_count	Integer	参数解释： 副本数。 取值范围： 不涉及。
inquiry_spec_code	String	参数解释： 产品规格编码，该参数为服务内部参数，用户无需关注。 取值范围： 不涉及。
billing_factor	String	参数解释： 计费模式，该参数为服务内部参数，用户无需关注。 取值范围： 不涉及。

表 5-898 AttrsObject

参数	参数类型	描述
capacity	String	参数解释： 缓存容量（GB）。 取值范围： 0.125、0.25、0.5、1、2、4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024、2048 等。
name	String	参数解释： 额外信息名。 取值范围： • sharding_num：该规格实例支持的分片数。 • proxy_num：该规格Proxy实例支持的Proxy节点数量。如果不是Proxy实例，该参数为0。 • db_number：该规格实例的DB数量。 • max_memory：实际可使用的最大内存。 • max_connections：该规格支持的最大连接数。 • max_clients：该规格支持的最大客户端数，一般等于最大连接数。 • max_bandwidth：该规格支持的最大带宽。 • max_in_bandwidth：该规格支持的最大接入带宽，一般等于最大带宽。
value	String	参数解释： 额外信息值。 取值范围： 不涉及。

表 5-899 FlavorAzObject

参数	参数类型	描述
capacity	String	参数解释： 缓存容量数值。 取值范围： 0.125、0.25、0.5、1、2、4、8、16、24、32、48、64、96、128、192、256、384、512、768、1024、2048 等。
unit	String	参数解释： 缓存容量单位。 取值范围： GB
available_zones	Array of strings	参数解释： 可用区信息。 取值范围： 不涉及。
az_codes	Array of strings	参数解释： 有资源的可用区编码。 取值范围： 不涉及。

状态码：400

表 5-900 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-901 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-902 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-903 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-904 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/flavors?  
cache_mode={cache_mode}&engine={engine}&engine_version={engine_version}&cpu_type={cpu_type}&capa  
city={capacity}
```

响应示例

状态码：200

查询产品规格成功。

```
{  
  "flavors": [ {  
    "spec_code": "redis.single.au1.large.1",  
    "cloud_service_type_code": "hws.resource.type.dcs3",  
    "cloud_resource_type_code": "hws.resource.type.dcs3",  
    "cache_mode": "single",  
    "engine": "redis",  
    "engine_version": "4.0;5.0",  
    "product_type": "generic",  
    "cpu_type": "aarch64",  
    "storage_type": "DRAM",  
    "capacity": [ 1 ],  
    "billing_mode": [ "Hourly", "Monthly", "Yearly" ],  
    "tenant_ip_count": 1,  
    "pricing_type": "normal",  
    "is_dec": false,  
    "attrs": [ {  
      "capacity": "1",  
      "name": "max_memory",  
      "value": "1"  
    }, {  
      "capacity": "1",  
      "name": "max_connections",  
      "value": "10000"  
    }, {  
      "capacity": "1",
```



```
    "name": "sharding_num",  
    "value": "1"  
  }, {  
    "capacity": "1",  
    "name": "proxy_num",  
    "value": "0"  
  }, {  
    "capacity": "1",  
    "name": "db_number",  
    "value": "256"  
  }, {  
    "capacity": "1",  
    "name": "max_clients",  
    "value": "10000"  
  }, {  
    "capacity": "1",  
    "name": "max_bandwidth",  
    "value": "80"  
  } ],  
  "flavors_available_zones": [ {  
    "capacity": "1",  
    "unit": "GB",  
    "available_zones": [ "d539378ec1314c85b76fefa3f7071458" ],  
    "az_codes": [ "region01" ]  
  } ],  
  "replica_count": 1,  
  "inquiry_spec_code": "redis.single.au1.large.1",  
  "billing_factor": "Duration"  
} ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;  
import com.huaweicloud.sdk.dcs.v2.*;  
import com.huaweicloud.sdk.dcs.v2.model.*;  
  
public class ListFlavorsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        DcsClient client = DcsClient.newBuilder()  
            .withCredential(auth)
```

```
        .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
        .build();
ListFlavorsRequest request = new ListFlavorsRequest();
try {
    ListFlavorsResponse response = client.listFlavors(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListFlavorsRequest()
        response = client.list_flavors(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFlavorsRequest{}
    response, err := client.ListFlavors(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询产品规格成功。
400	非法请求。
401	认证信息非法
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.2 查询维护时间窗时间段

功能介绍

查询维护时间窗开始时间和结束时间。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/instances/maintain-windows

请求参数

无

响应参数

状态码：200

表 5-905 响应 Body 参数

参数	参数类型	描述
maintain_windows	Array of MaintainWindowsEntity objects	参数解释： 支持的维护时间窗列表。 取值范围： 不涉及。

表 5-906 MaintainWindowsEntity

参数	参数类型	描述
seq	Integer	参数解释： 序号。 取值范围： 不涉及。
default	Boolean	参数解释： 是否为默认时间段。 取值范围： - true - false

参数	参数类型	描述
begin	String	参数解释： 维护时间窗开始时间，时间为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。
end	String	参数解释： 维护时间窗结束时间，时间为UTC时间，格式为HH:mm:ss。 取值范围： 不涉及。

状态码：500

表 5-907 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/instances/maintain-windows
```

响应示例

状态码：200

查询维护时间窗时间段成功。

```
{  
  "maintain_windows" : [ {
```

```

    "default" : true,
    "end" : "19",
    "begin" : "18",
    "seq" : 1
  }, {
    "default" : false,
    "end" : "20",
    "begin" : "19",
    "seq" : 2
  }, {
    "default" : false,
    "end" : "21",
    "begin" : "20",
    "seq" : 3
  }, {
    "default" : false,
    "end" : "22",
    "begin" : "21",
    "seq" : 4
  }, {
    "default" : false,
    "end" : "23",
    "begin" : "22",
    "seq" : 5
  }, {
    "default" : false,
    "end" : "00",
    "begin" : "23",
    "seq" : 6
  }, {
    "default" : false,
    "end" : "01",
    "begin" : "00",
    "seq" : 1
  }, {
    "default" : false,
    "end" : "02",
    "begin" : "01",
    "seq" : 2
  }, {
    "default" : false,
    "end" : "03",
    "begin" : "02",
    "seq" : 3
  }, {
    "default" : false,
    "end" : "04",
    "begin" : "03",
    "seq" : 3
  }, {
    "default" : false,
    "end" : "05",
    "begin" : "04",
    "seq" : 4
  }, {
    "default" : false,
    "end" : "06",
    "begin" : "05",
    "seq" : 5
  }, {
    "default" : false,
    "end" : "07",
    "begin" : "06",
    "seq" : 6
  }, {
    "default" : false,
    "end" : "08",
    "begin" : "07",
    "seq" : 1
  }

```

```
}, {
  "default" : true,
  "end" : "09",
  "begin" : "08",
  "seq" : 2
}, {
  "default" : false,
  "end" : "10",
  "begin" : "09",
  "seq" : 3
}, {
  "default" : false,
  "end" : "11",
  "begin" : "10",
  "seq" : 4
}, {
  "default" : false,
  "end" : "12",
  "begin" : "11",
  "seq" : 5
}, {
  "default" : false,
  "end" : "13",
  "begin" : "12",
  "seq" : 6
}, {
  "default" : false,
  "end" : "14",
  "begin" : "13",
  "seq" : 1
}, {
  "default" : true,
  "end" : "15",
  "begin" : "14",
  "seq" : 2
}, {
  "default" : false,
  "end" : "16",
  "begin" : "15",
  "seq" : 3
}, {
  "default" : false,
  "end" : "17",
  "begin" : "16",
  "seq" : 4
}, {
  "default" : false,
  "end" : "18",
  "begin" : "17",
  "seq" : 5
}]
}
```

状态码

状态码	描述
200	查询维护时间窗时间段成功。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.3 查询租户配额

功能介绍

查询租户默认可以创建的实例数和总内存的配额限制，以及可以申请配额的最大值和最小值。不同的租户在不同的区域配额可能不同。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/quota

表 5-908 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-909 响应 Body 参数

参数	参数类型	描述
quotas	quotas object	参数解释： 配额信息。 取值范围： 不涉及。

表 5-910 quotas

参数	参数类型	描述
resources	Array of Resources objects	参数解释： 配额列表。 取值范围： 不涉及。

表 5-911 Resources

参数	参数类型	描述
unit	String	参数解释： 资源的计数单位。 - 当type为instance时，无单位。 - 当type为ram时，单位为GB。 取值范围： 不涉及。
min	Integer	参数解释： - 当type为instance时，表示可申请实例配额的最小值。 - 当type为ram时，表示可申请内存配额的最小值。 取值范围： 不涉及。
max	Integer	参数解释： - 当type为instance时，表示可申请实例配额的最大值。 - 当type为ram时，表示可申请内存配额的最大值。 取值范围： 不涉及。
quota	Integer	参数解释： 可以创建的实例最大数和总内存的配额限制。 取值范围： 不涉及。

参数	参数类型	描述
used	Integer	参数解释： 已创建的实例个数和已使用的内存配额。 取值范围： 不涉及。
type	String	参数解释： 配额类型。 取值范围： - instance：表示实例配额。 - ram：表示内存配额。

状态码：400

表 5-912 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-913 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/quota
```

响应示例

状态码：200

查询租户配额成功。

```
{
  "quotas": {
    "resources": [ {
      "unit": { },
      "min": 1,
      "max": 10,
      "quota": 10,
      "used": 3,
      "type": "instance"
    }, {
      "unit": "GB",
      "min": 1,
      "max": 800,
      "quota": 800,
      "used": 22,
      "type": "ram"
    }
  ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ShowQuotaOfTenantSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowQuotaOfTenantRequest request = new ShowQuotaOfTenantRequest();
        try {
            ShowQuotaOfTenantResponse response = client.showQuotaOfTenant(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowQuotaOfTenantRequest()
    response = client.show_quota_of_tenant(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowQuotaOfTenantRequest{}
    response, err := client.ShowQuotaOfTenant(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询租户配额成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.4 查询单个主维度下子维度监控对象列表

功能介绍

查询主维度下子维度监控对象列表，当前支持子维度的主维度ID有dcs_instance_id。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/dms/monitored-objects/{instance_id}

表 5-914 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 主维度对象ID，与DCS实例列表中实例ID相同。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-915 Query 参数

参数	是否必选	参数类型	描述
dim_name	是	String	参数解释： 主维度ID，当前支持子维度的主维度ID的有dcs_instance_id。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-916 响应 Body 参数

参数	参数类型	描述
router	Array of strings	参数解释： 当前查询维度路由。如果是主维度，则数组中是自身ID。 取值范围： 不涉及。

参数	参数类型	描述
children	Array of DimChild objects	参数解释: 当前查询维度子维度对象列表。当前只有维度为dcs_instance_id时才有值。 取值范围: - Proxy集群有两个子维度, 分别为dcs_cluster_redis_node和dcs_cluster_proxy_node。 - Cluster集群有一个子维度dcs_cluster_proxy_node。
instances	Array of InstancesMonitoredObject objects	参数解释: 当前查询维度监控对象列表。 取值范围: 不涉及。
dcs_cluster_redis_node	Array of ClusterRedisNodeMonitoredObject objects	参数解释: Proxy集群或Cluster集群时才存在, 表示集群数据节点维度的监控对象列表。字段名称与children的子维度对象名称相同。 取值范围: 不涉及。
dcs_cluster_proxy_node	Array of ProxyNodeMonitoredObject objects	参数解释: Redis 3.0 Proxy集群时才存在, 表示集群Proxy节点维度的监控对象列表。字段名称与children的子维度对象名称相同。 取值范围: 不涉及。
dcs_cluster_proxy2_node	Array of Proxy2NodeMonitoredObject objects	参数解释: Redis 4.0和5.0的Proxy集群时才存在, 表示集群Proxy节点维度的监控对象列表。字段名称与children的子维度对象名称相同。 取值范围: 不涉及。
total	Integer	参数解释: 主维度监控对象的总数。 取值范围: 不涉及。

表 5-917 DimChild

参数	参数类型	描述
dim_name	String	参数解释： 维度名称。 取值范围： • dcs_instance_id • dcs_cluster_redis_node • dcs_cluster_proxy_node • dcs_memcached_instance_id
dim_route	String	参数解释： 维度的路由，结构为主维度名称，当前维度名称，例如： dim_name字段为 dcs_cluster_redis_node时，这个字段的值为dcs_instance_id， dcs_cluster_redis_node。 取值范围： 不涉及。

表 5-918 InstancesMonitoredObject

参数	参数类型	描述
dcs_instance_id	String	参数解释： 测量对象ID，即实例的ID。 取值范围： 不涉及。
name	String	参数解释： 测量对象名称，即实例名称。 取值范围： 不涉及。
status	String	参数解释： 测量对象状态，即实例状态。 取值范围： 不涉及。

表 5-919 ClusterRedisNodeMonitoredObject

参数	参数类型	描述
dc_instance_id	String	参数解释： 测量对象ID，即节点的ID。 取值范围： 不涉及。
name	String	参数解释： 测量对象名称，即节点IP。 取值范围： 不涉及。
dc_cluster_redis_node	String	参数解释： 维度dc_cluster_redis_node的测量对象的ID。 取值范围： 不涉及。
status	String	参数解释： 测量对象状态，即节点状态。 取值范围： 不涉及。

表 5-920 ProxyNodeMonitoredObject

参数	参数类型	描述
dc_instance_id	String	参数解释： 测量对象ID，即节点的ID。 取值范围： 不涉及。
name	String	参数解释： 测量对象名称，即节点IP。 取值范围： 不涉及。
dc_cluster_proxy_node	String	参数解释： 维度dc_cluster_proxy_node的测量对象的ID。 取值范围： 不涉及。

参数	参数类型	描述
status	String	参数解释： 测量对象状态，即节点状态。 取值范围： 不涉及。

表 5-921 Proxy2NodeMonitoredObject

参数	参数类型	描述
dcs_instance_id	String	参数解释： 测量对象ID，即节点的ID。 取值范围： 不涉及。
name	String	参数解释： 测量对象名称，即节点IP。 取值范围： 不涉及。
dcs_cluster_proxy2_node	String	参数解释： 维度dcs_cluster_proxy2_node 的测量对象的ID。 取值范围： 不涉及。
status	String	参数解释： 测量对象状态，即节点状态。 取值范围： 不涉及。

状态码：500

表 5-922 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/dims/monitored-objects/{instance_id}?dim_name={dim_name}
```

响应示例

状态码：200
查询主维度监控对象成功。

```
{
  "router": [ "dcs_instance_id" ],
  "total": 1,
  "children": [ {
    "dim_name": "dcs_cluster_redis_node",
    "dim_route": "dcs_instance_id,dcs_cluster_redis_node"
  } ],
  "instances": [ {
    "name": "dcs-test001",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19"
  } ],
  "dcs_cluster_redis_node": [ {
    "name": "(master)192.168.2.145",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",
    "dcs_cluster_redis_node": "b1f4aa9e4dae50888e58c9caecdfc108"
  }, {
    "name": "(replica)192.168.2.199",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",
    "dcs_cluster_redis_node": "e0e10e489a73487147928167396474bc"
  }, {
    "name": "(master)192.168.2.243",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",
    "dcs_cluster_redis_node": "e797c3ba8bee782e25cdd4a90eb00517"
  }, {
    "name": "(replica)192.168.2.164",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",
    "dcs_cluster_redis_node": "d810fb44f9f7a359e000cf277a824c43"
  }, {
    "name": "(master)192.168.2.95",
    "status": "RUNNING",
    "dcs_instance_id": "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",
    "dcs_cluster_redis_node": "f25c37a4654d50b07e629fc2adfa60f"
  } ]
}
```

```
}, {  
  "name" : "(replica)192.168.2.51",  
  "status" : "RUNNING",  
  "dcs_instance_id" : "03ca9da0-1104-40a5-b49d-5ef8e41bfd19",  
  "dcs_cluster_redis_node" : "da5149a20dc7caf35587e4d2433fe452"  
}]  
}
```

状态码

状态码	描述
200	查询主维度监控对象成功。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.5 查询主维度信息列表

功能介绍

查询主维度对象列表，主维度ID当前支持dcs_instance_id，dcs_memcached_instance_id。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/dims/monitored-objects

表 5-923 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-924 Query 参数

参数	是否必选	参数类型	描述
dim_name	是	String	参数解释： 主维度ID。 约束限制： 不涉及。 取值范围： 当前支持dcs_instance_id， dcs_memcached_instance_id。 默认取值： 不涉及。
offset	否	Integer	参数解释： 偏移量，表示生成的列表从此偏移量开始显示，例如偏移量为3时，生成的列表从第4条开始。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 1-1000 默认取值： 10

请求参数

无

响应参数

状态码： 200

表 5-925 响应 Body 参数

参数	参数类型	描述
router	Array of strings	参数解释： 当前查询维度路由。如果是主维度，则数组中是自身ID。 取值范围： 不涉及。
children	Array of DimChild objects	参数解释： 当前查询维度子维度对象列表。当前只有维度为dcs_instance_id时才有值。 取值范围： - Proxy集群有两个子维度，分别为： dcs_cluster_redis_node和 dcs_cluster_proxy_node。 - Cluster集群有一个子维度： dcs_cluster_proxy_node。
instances	Array of InstancesMonitoredObject objects	参数解释： 当前查询维度监控对象列表。 取值范围： 不涉及。
total	Integer	参数解释： 主维度监控对象的总数。 取值范围： 不涉及。

表 5-926 DimChild

参数	参数类型	描述
dim_name	String	参数解释： 维度名称。 取值范围： - dcs_instance_id - dcs_cluster_redis_node - dcs_cluster_proxy_node - dcs_memcached_instance_id

参数	参数类型	描述
dim_route	String	参数解释： 维度的路由，结构为主维度名称，当前维度名称，例如： dim_name字段为 dcs_cluster_redis_node时，这个字段的值为dcs_instance_id， dcs_cluster_redis_node。 取值范围： 不涉及。

表 5-927 InstancesMonitoredObject

参数	参数类型	描述
dcs_instance_id	String	参数解释： 测量对象ID，即实例的ID。 取值范围： 不涉及。
name	String	参数解释： 测量对象名称，即实例名称。 取值范围： 不涉及。
status	String	参数解释： 测量对象状态，即实例状态。 取值范围： 不涉及。

状态码： 500

表 5-928 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/dms/monitored-objects?dim_name={dim_name}
```

响应示例

状态码：200

查询主维度监控列表成功。

```
{
  "router" : [ "dcs_instance_id" ],
  "total" : 3,
  "children" : [ {
    "dim_name" : "dcs_cluster_redis_node",
    "dim_route" : "dcs_instance_id,dcs_cluster_redis_node"
  }, {
    "dim_name" : "dcs_cluster_proxy_node",
    "dim_route" : "dcs_instance_id,dcs_cluster_proxy_node"
  } ],
  "instances" : [ {
    "name" : "dcs-redis-single-node",
    "status" : "RUNNING",
    "dcs_instance_id" : "fe909c47-8990-44a0-9154-d0a1e95e78fe"
  }, {
    "name" : "dcs-redis-master-standby",
    "status" : "RUNNING",
    "dcs_instance_id" : "877e5ae3-482e-4c38-88a0-030a0fa6f399"
  }, {
    "name" : "dcs-proxy-cluster",
    "status" : "RUNNING",
    "dcs_instance_id" : "448ee851-1366-47f2-913a-e21032e690c4"
  } ]
}
```

状态码

状态码	描述
200	查询主维度监控列表成功。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.6 查询可用区信息

功能介绍

查询所在局点的可用区信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/available-zones

请求参数

无

响应参数

状态码：200

表 5-929 响应 Body 参数

参数	参数类型	描述
region_id	String	参数解释： 区域ID。 取值范围： 不涉及。
available_zones	Array of AvailableZones objects	参数解释： 可用分区数组。 取值范围： 不涉及。

表 5-930 AvailableZones

参数	参数类型	描述
code	String	参数解释： 可用区编码。 取值范围： 不涉及。

参数	参数类型	描述
port	String	参数解释： 可用区端口号。 取值范围： 不涉及。
name	String	参数解释： 可用区名称。 取值范围： 不涉及。
id	String	参数解释： 可用区ID。 取值范围： 不涉及。
resource_availability	String	参数解释： 分区上是否有可用资源。 取值范围： • true：有可用资源。 • false：没有可用资源。

请求示例

```
GET https://{dcs_endpoint}/v2/available-zones
```

响应示例

状态码：200

查询成功。

```
{
  "region_id": "region00",
  "available_zones": [ {
    "code": "region01",
    "port": "8403",
    "name": "可用区1",
    "id": "effdcbc7d4d64a02aa1fa26b42f56533",
    "resource_availability": "true"
  }, {
    "code": "region02",
    "port": "8404",
    "name": "可用区2",
    "id": "a0865121f83b41cbafce65930a22a6e8",
    "resource_availability": "true"
  }, {
    "code": "region03",
    "port": "8408",
    "name": "可用区3",
    "id": "2dcb154ac2724a6d92e9bcc859657c1e",
    "resource_availability": "true"
  }
]
```

状态码

状态码	描述
200	查询成功。

错误码

请参见[错误码](#)。

5.15.7 执行 web-cli 命令

功能介绍

登入web-cli，执行redis命令。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/webcli/command

表 5-931 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-932 请求 Body 参数

参数	是否必选	参数类型	描述
client_id	是	String	客户端ID，可通过 登录WebCli 接口的返回体获取。
command	否	String	命令。
database	否	Integer	数据库编号。

响应参数

状态码：200

表 5-933 响应 Body 参数

参数	参数类型	描述
response	String	返回信息。

状态码：400

表 5-934 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-935 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。

参数	参数类型	描述
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

请求示例

```
post https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/webcli/command
{
  "client_id" : "cf6135014f82d1de212d4a2174fb149c8756193347864ff7",
  "database" : 0,
  "command" : "scan 0"
}
```

响应示例

状态码：200

返回成功。

```
{
  "response" : "key1\nkey2"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ExecuteCommandMobilizationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
```

```
.withAk(ak)
.withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ExecuteCommandMobilizationRequest request = new ExecuteCommandMobilizationRequest();
request.withInstanceId("{instance_id}");
ExecuteCommandRequestBody body = new ExecuteCommandRequestBody();
body.withDatabase(0);
body.withCommand("scan 0");
body.withClientId("cf6135014f82d1de212d4a2174fb149c8756193347864ff7");
request.withBody(body);
try {
    ExecuteCommandMobilizationResponse response = client.executeCommandMobilization(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ExecuteCommandMobilizationRequest()
        request.instance_id = "{instance_id}"
        request.body = ExecuteCommandRequestBody(
            database=0,
            command="scan 0",
            client_id="cf6135014f82d1de212d4a2174fb149c8756193347864ff7"
        )
        response = client.execute_command_mobilization(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteCommandMobilizationRequest{}
    request.InstanceId = "{instance_id}"
    databaseExecuteCommandRequestBody := int32(0)
    commandExecuteCommandRequestBody := "scan 0"
    clientIdExecuteCommandRequestBody := "cf6135014f82d1de212d4a2174fb149c8756193347864ff7"
    request.Body = &model.ExecuteCommandRequestBody{
        Database: &databaseExecuteCommandRequestBody,
        Command: &commandExecuteCommandRequestBody,
        ClientId: &clientIdExecuteCommandRequestBody,
    }
    response, err := client.ExecuteCommandMobilization(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	返回成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.8 登出 WebCli

功能介绍

登出WebCli。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/webcli/logout

表 5-936 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-937 请求 Body 参数

参数	是否必选	参数类型	描述
client_id	否	String	客户端ID。

响应参数

状态码：204

登出WebCli成功。

状态码：400

表 5-938 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-939 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：403

表 5-940 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-941 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-942 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/webcli/logout
{
  "client_id" : "45934af80681efd5282554d77cae150760df78a9b342568c"
}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class LogoffWebCliSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        LogoffWebCliRequest request = new LogoffWebCliRequest();
        request.withInstanceId("{instance_id}");
        LogoutWebCliBody body = new LogoutWebCliBody();
        body.withClientId("45934af80681efd5282554d77cae150760df78a9b342568c");
        request.withBody(body);
        try {
            LogoffWebCliResponse response = client.logoffWebCli(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
    request = LogoffWebCliRequest()
    request.instance_id = "{instance_id}"
    request.body = LogoutWebCliBody(
        client_id="45934af80681efd5282554d77cae150760df78a9b342568c"
    )
    response = client.logoff_web_cli(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.LogoffWebCliRequest{}
    request.InstanceId = "{instance_id}"
    clientIdLogoutWebCliBody := "45934af80681efd5282554d77cae150760df78a9b342568c"
    request.Body = &model.LogoutWebCliBody{
        ClientId: &clientIdLogoutWebCliBody,
    }
    response, err := client.LogoffWebCli(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	登出WebCli成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.15.9 登录 WebCli

功能介绍

登录WebCli。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/webcli/auth

表 5-943 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-944 请求 Body 参数

参数	是否必选	参数类型	描述
password	否	String	登录Redis密码。

响应参数

状态码：200

表 5-945 响应 Body 参数

参数	参数类型	描述
client_id	String	客户端ID。
databases	Integer	DB数量。

状态码：400

表 5-946 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为null）。 取值范围： 不涉及。

状态码：401

表 5-947 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。

参数	参数类型	描述
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-948 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-949 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-950 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/webcli/auth
{
```

```
{
  "password" : "xxxxxx"
}
```

响应示例

状态码：200

登录WebCli成功。

```
{
  "client_id" : "238d8f22ccbc1f497482d5c8c87df023de14ea3eae55dc8",
  "databases" : 256
}
```

状态码：400

非法请求。

```
{
  "error_code" : "111400063",
  "error_msg" : "Invalid {0} parameter in the request."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class LoginWebCliSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        LoginWebCliRequest request = new LoginWebCliRequest();
        request.withInstanceId("{instance_id}");
        LoginWebCliBody body = new LoginWebCliBody();
        body.withPassword("xxxxxx");
        request.withBody(body);
    }
}
```

```
try {
    LoginWebCliResponse response = client.loginWebCli(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = LoginWebCliRequest()
        request.instance_id = "{instance_id}"
        request.body = LoginWebCliBody(
            password="xxxxxx"
        )
        response = client.login_web_cli(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.LoginWebCliRequest{}
    request.InstanceId = "{instance_id}"
    passwordLoginWebCliBody:= "xxxxxx"
    request.Body = &model.LoginWebCliBody{
        Password: &passwordLoginWebCliBody,
    }
    response, err := client.LoginWebCli(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	登录WebCli成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.16 会话管理

5.16.1 下发查询会话列表任务

功能介绍

下发查询会话列表任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/clients

表 5-951 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-952 请求 Body 参数

参数	是否必选	参数类型	描述
node_id	是	String	节点ID。
clean_cache	否	Boolean	是否重新查询并保存会话列表。

响应参数

状态码：200

表 5-953 响应 Body 参数

参数	参数类型	描述
job_id	String	下发查询会话列表任务ID。

状态码：400

表 5-954 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：401

表 5-955 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：403

表 5-956 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：404

表 5-957 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：500

表 5-958 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/clients
{
  "node_id" : "8f7ee181-88e2-49b7-a4bf-81f18f93a135",
  "clean_cache" : true
}
```

响应示例

状态码：200

下发查询会话列表任务成功。

```
{
  "job_id" : "8abf6e1e824e204401824e3e621b0042"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4957",
  "error_msg" : "This operation is not allowed when the task is in the current state."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ScanClientsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}
```



```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

DcsClient client = DcsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
    .build();
ScanClientsRequest request = new ScanClientsRequest();
request.withInstanceId("{instance_id}");
ScanClientsRequestBody body = new ScanClientsRequestBody();
body.withCleanCache(true);
body.withNodeId("8f7ee181-88e2-49b7-a4bf-81f18f93a135");
request.withBody(body);
try {
    ScanClientsResponse response = client.scanClients(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ScanClientsRequest()
        request.instance_id = "{instance_id}"
        request.body = ScanClientsRequestBody()
```

```
        clean_cache=True,
        node_id="8f7ee181-88e2-49b7-a4bf-81f18f93a135"
    )
    response = client.scan_clients(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ScanClientsRequest{}
    request.InstanceId = "{instance_id}"
    cleanCacheScanClientsRequestBody := true
    request.Body = &model.ScanClientsRequestBody{
        CleanCache: &cleanCacheScanClientsRequestBody,
        NodeId: "8f7ee181-88e2-49b7-a4bf-81f18f93a135",
    }
    response, err := client.ScanClients(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下发查询会话列表任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.16.2 获取会话列表

功能介绍

获取会话列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/clients

表 5-959 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

表 5-960 Query 参数

参数	是否必选	参数类型	描述
node_id	是	String	节点ID。如果是读写分离、proxy集群实例请使用proxy节点node_id，如果是单机、主备、cluster集群实例请使用数据节点的node_id。
offset	否	Integer	偏移量，表示从此偏移量开始查询，偏移量大于等于0，默认为0。
limit	否	Integer	每页显示条数，最小值为1，最大值为1000，默认值为10。
addr	否	String	按客户端连接地址过滤。
sort	否	String	排序字段，指定会话列表ClientInfo中的一个值进行排序。
order	否	String	排序方式，asc：正序排序，desc：倒序排序。

请求参数

无

响应参数

状态码：200

表 5-961 响应 Body 参数

参数	参数类型	描述
time	String	数据更新时间。
clients	Array of ClientInfo objects	会话列表。
count	Integer	会话总数。

表 5-962 ClientInfo

参数	参数类型	描述
id	String	客户端ID。
addr	String	客户端所在的地址和端口。

参数	参数类型	描述
fd	String	容器内所使用的文件描述符。
name	String	客户端的名称。
cmd	String	最近一次执行的命令。
age	Integer	已连接时长（单位：秒）。
idle	Integer	空闲时长（单位：秒）。
db	String	该客户端正在使用的数据库ID。
flags	String	客户端标志。默认未设置，值为N。
sub	Integer	已订阅频道的数量。
psub	Integer	已订阅模式的数量。
multi	Integer	在事务中被执行的命令数量。
qbuf	Integer	查询缓冲区的长度（单位为字节，0表示没有分配查询缓冲区）。
qbuf_free	Integer	查询缓冲区剩余空间的长度（单位为字节，0表示没有剩余空间）。
obl	Integer	输出缓冲区的长度（单位为字节，0表示没有分配输出缓冲区）。
oll	Integer	输出列表包含的对象数量（当输出缓冲区没有剩余空间时，命令回复会以字符串对象的形式被入队到这个队列里）
omem	Integer	输出缓冲区和输出列表占用的内存总量。
events	String	文件操作事件，r:读;w:写。
network	String	客户端所使用的网络类型。
peer	String	client地址和端口。
user	String	客户端用户。

状态码：400

表 5-963 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：401

表 5-964 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：403

表 5-965 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：404

表 5-966 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：500

表 5-967 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/clients?node_id=8f7ee181-88e2-49b7-a4bf-81f18f93a135
```

响应示例

状态码：200

查询会话管理列表成功。

```
{
  "time" : "2023-06-15T03:10:01.332Z",
  "clients" : [ {
    "id" : "11677",
    "addr" : "198.19.131.14:1637",
    "name" : "dcs",
    "cmd" : "info",
    "age" : 376,
    "idle" : 376,
    "flags" : "N",
    "fd" : "16",
    "db" : "0",
    "sub" : 0,
    "psub" : 0,
    "multi" : 1,
    "qbuf" : 0,
    "qbuf_free" : 4,
    "obl" : 0,
    "oll" : 0,
    "omem" : 0,
    "events" : "r",
    "network" : "vpc",
    "peer" : "198.198.0.121:4058",
    "user" : "dcs"
  } ],
  "count" : 1
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4957",
  "error_msg" : "This operation is not allowed when the task is in the current state."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListClientsSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    DcsClient client = DcsClient.newBuilder()
        .withCredential(auth)
        .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
        .build();
    ListClientsRequest request = new ListClientsRequest();
    request.withInstanceId("{instance_id}");
    try {
        ListClientsResponse response = client.listClients(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListClientsRequest()
        request.instance_id = "{instance_id}"
```



```
response = client.list_clients(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClientsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListClients(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询会话管理列表成功。
400	非法请求。

状态码	描述
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.16.3 kill 指定的会话

功能介绍

kill指定的会话。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/clients/kill

表 5-968 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

表 5-969 请求 Body 参数

参数	是否必选	参数类型	描述
node_id	是	String	节点ID。
client_addrs	是	Array of strings	要kill的会话地址列表，取值为会话信息中peer的地址和端口。

响应参数

状态码：204

kill指定的会话成功。

状态码：400

表 5-970 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：401

表 5-971 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：403

表 5-972 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：404

表 5-973 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。

参数	参数类型	描述
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：500

表 5-974 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

请求示例

```
POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/clients/kill
{
  "node_id" : "8f7ee181-88e2-49b7-a4bf-81f18f93a135",
  "client_addrs" : [ "10.63.238.158:2048", "10.63.238.159:3058" ]
}
```

响应示例

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4957",
  "error_msg" : "This operation is not allowed when the task is in the current state."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

import java.util.List;
import java.util.ArrayList;
```

```
public class HangUpClientsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        HangUpClientsRequest request = new HangUpClientsRequest();
        request.withInstanceId("{instance_id}");
        HangUpClientsRequestBody body = new HangUpClientsRequestBody();
        List<String> listbodyClientAddrs = new ArrayList<>();
        listbodyClientAddrs.add("10.63.238.158:2048");
        listbodyClientAddrs.add("10.63.238.159:3058");
        body.withClientAddrs(listbodyClientAddrs);
        body.withNodeId("8f7ee181-88e2-49b7-a4bf-81f18f93a135");
        request.withBody(body);
        try {
            HangUpClientsResponse response = client.hangUpClients(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = HangUpClientsRequest()
    request.instance_id = "{instance_id}"
    listClientAdrsbody = [
        "10.63.238.158:2048",
        "10.63.238.159:3058"
    ]
    request.body = HangUpClientsRequestBody(
        client_addrs=listClientAdrsbody,
        node_id="8f7ee181-88e2-49b7-a4bf-81f18f93a135"
    )
    response = client.hang_up_clients(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.HangUpClientsRequest{}
    request.InstanceId = "{instance_id}"
    var listClientAdrsbody = []string{
        "10.63.238.158:2048",
        "10.63.238.159:3058",
    }
    request.Body = &model.HangUpClientsRequestBody{
        ClientAdrs: listClientAdrsbody,
        NodeId: "8f7ee181-88e2-49b7-a4bf-81f18f93a135",
    }
}
```

```
response, err := client.HangUpClients(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	kill指定的会话成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.16.4 下发 kill 指定节点或实例的全部会话任务

功能介绍

下发kill指定节点或实例的全部会话任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/clients/kill-all

表 5-975 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。

参数	是否必选	参数类型	描述
instance_id	是	String	实例ID。

请求参数

表 5-976 请求 Body 参数

参数	是否必选	参数类型	描述
node_id	否	String	指定要kill全部会话的节点ID，删除所有节点为false时必填。
kill_all_nodes	否	Boolean	true: Kill实例全部节点的会话。 false: kill指定节点的全部会话。

响应参数

状态码：200

表 5-977 响应 Body 参数

参数	参数类型	描述
job_id	String	下发kill指定节点或实例的任务ID。

状态码：400

表 5-978 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：401

表 5-979 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：403

表 5-980 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：404

表 5-981 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

状态码：500

表 5-982 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。
error_code	String	错误码。
error_ext_msg	String	扩展错误信息。

请求示例

```
{
  "node_id" : "8f7ee181-88e2-49b7-a4bf-81f18f93a135",
}
```

```
"kill_all_nodes" : true
}
```

响应示例

状态码：200

下发kill指定节点或实例的全部会话任务成功。

```
{
  "job_id" : "8abf6e1e824dfcb201824e02ee9a0015"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4957",
  "error_msg" : "This operation is not allowed when the task is in the current state."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class HangUpKillAllClientsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        HangUpKillAllClientsRequest request = new HangUpKillAllClientsRequest();
        request.withInstanceId("{instance_id}");
        HangUpKillAllClientsRequestBody body = new HangUpKillAllClientsRequestBody();
        body.withKillAllNodes(true);
        body.withNodeId("8f7ee181-88e2-49b7-a4bf-81f18f93a135");
        request.withBody(body);
```

```
try {
    HangUpKillAllClientsResponse response = client.hangUpKillAllClients(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = HangUpKillAllClientsRequest()
        request.instance_id = "{instance_id}"
        request.body = HangUpKillAllClientsRequestBody(
            kill_all_nodes=True,
            node_id="8f7ee181-88e2-49b7-a4bf-81f18f93a135"
        )
        response = client.hang_up_kill_all_clients(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.HangUpKillAllClientsRequest{}
    request.InstanceId = "{instance_id}"
    killAllNodesHangUpKillAllClientsRequestBody:= true
    nodeIdHangUpKillAllClientsRequestBody:= "8f7ee181-88e2-49b7-a4bf-81f18f93a135"
    request.Body = &model.HangUpKillAllClientsRequestBody{
        KillAllNodes: &killAllNodesHangUpKillAllClientsRequestBody,
        NodeId: &nodeIdHangUpKillAllClientsRequestBody,
    }
    response, err := client.HangUpKillAllClients(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	下发kill指定节点或实例的全部会话任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.17 离线全量 key 分析

5.17.1 创建离线全量 key 分析任务

功能介绍

创建离线全量key分析任务。离线全量key分析用于分析实例指定节点备份文件中的TOP100大key，每种数据类型前缀数量TOP50的key和每种数据类型key的内存占用和数量的分布情况。仅Redis 4.0、5.0、6.0版本及Redis企业版实例支持。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/instances/{instance_id}/offline/key-analysis

表 5-983 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 5-984 请求 Body 参数

参数	是否必选	参数类型	描述
node_id	是	String	参数解释： 实例节点ID，可通过DCS控制台进入实例的节点管理界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
backup_id	否	String	参数解释： 实例备份ID，可通过DCS控制台进入实例的备份与恢复界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

响应参数

状态码：200

表 5-985 响应 Body 参数

参数	参数类型	描述
task_id	String	参数解释： 离线全量key分析任务ID。 取值范围： 不涉及。

状态码：400

表 5-986 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-987 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-988 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-989 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-990 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

- POST https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/offline/key-analysis

```
{
  "node_id" : "ff8080828fdd5aef018fe2218a886105"
}
```
- ```
{
 "node_id" : "ff8080828fdd5aef018fe2218a886105",
 "backup_id" : "c24e2ef4-23bd-47ef-8b1f-8e3f0492c906"
}
```

响应示例

状态码：200

创建离线全量key分析任务成功。

```
{
 "task_id" : "aa735b6f-3d70-44e3-9771-66c6fae7459a"
}
```

状态码：400

非法请求。

```
{
 "error_code" : "DCS.4230",
 "error_msg" : "Does not support offline key analysis."
}
```

SDK 代码示例

SDK代码示例如下。

## Java

- ```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class CreateOfflineKeyAnalysisSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateOfflineKeyAnalysisRequest request = new CreateOfflineKeyAnalysisRequest();
        request.withInstanceId("{instance_id}");
        CreateOfflineKeyAnalysisRequestBody body = new CreateOfflineKeyAnalysisRequestBody();
        body.withNodeId("ff8080828fdd5aef018fe2218a886105");
        request.withBody(body);
        try {
            CreateOfflineKeyAnalysisResponse response = client.createOfflineKeyAnalysis(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```
- ```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```

```
public class CreateOfflineKeyAnalysisSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");
 String projectId = "{project_id}";

 ICredential auth = new BasicCredentials()
 .withProjectId(projectId)
 .withAk(ak)
 .withSk(sk);

 DcsClient client = DcsClient.newBuilder()
 .withCredential(auth)
 .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateOfflineKeyAnalysisRequest request = new CreateOfflineKeyAnalysisRequest();
 request.withInstanceId("{instance_id}");
 CreateOfflineKeyAnalysisRequestBody body = new CreateOfflineKeyAnalysisRequestBody();
 body.withBackupId("c24e2ef4-23bd-47ef-8b1f-8e3f0492c906");
 body.withNodeId("ff8080828fdd5aef018fe2218a886105");
 request.withBody(body);
 try {
 CreateOfflineKeyAnalysisResponse response = client.createOfflineKeyAnalysis(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

- ```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(DcsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = CreateOfflineKeyAnalysisRequest()
    request.instance_id = "{instance_id}"
    request.body = CreateOfflineKeyAnalysisRequestBody(
        node_id="ff8080828fdd5aef018fe2218a886105"
    )
    response = client.create_offline_key_analysis(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- # coding: utf-8

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateOfflineKeyAnalysisRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateOfflineKeyAnalysisRequestBody(
            backup_id="c24e2ef4-23bd-47ef-8b1f-8e3f0492c906",
            node_id="ff8080828fdd5aef018fe2218a886105"
        )
        response = client.create_offline_key_analysis(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- package main

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateOfflineKeyAnalysisRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.CreateOfflineKeyAnalysisRequestBody{
        NodeId: "ff8080828fdd5aef018fe2218a886105",
    }
    response, err := client.CreateOfflineKeyAnalysis(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- package main

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.CreateOfflineKeyAnalysisRequest{}
request.InstanceId = "{instance_id}"
backupIdCreateOfflineKeyAnalysisRequestBody := "c24e2ef4-23bd-47ef-8b1f-8e3f0492c906"
request.Body = &model.CreateOfflineKeyAnalysisRequestBody{
    BackupId: &backupIdCreateOfflineKeyAnalysisRequestBody,
    NodeId: "ff8080828fdd5aef018fe2218a886105",
}
response, err := client.CreateOfflineKeyAnalysis(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	创建离线全量key分析任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.17.2 查询离线全量 key 分析任务列表

功能介绍

查询离线全量key分析任务列表，支持Redis4.0、5.0、6.0版本及Redis企业版。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/offline/key-analysis

表 5-991 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 5-992 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 偏移量，表示从此偏移量开始查询。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 0

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释： 每页显示的条目数量。 约束限制： 不涉及。 取值范围： 大于等于0。 默认取值： 10
status	否	String	参数解释： 离线分析的任务状态。 约束限制： 不涉及。 取值范围： waiting：任务等待中。 running：任务进行中。 success：任务执行成功。 failed：任务执行失败。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-993 响应 Body 参数

参数	参数类型	描述
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
total	Integer	参数解释： 离线全量key分析的总数。 取值范围： 不涉及。

参数	参数类型	描述
records	Array of OfflineKeyAnalysisRecord objects	参数解释： 离线全量key分析记录列表。

表 5-994 OfflineKeyAnalysisRecord

参数	参数类型	描述
id	String	参数解释： 任务执行记录ID（此ID对应“查询离线全量key分析详情”参数中的任务ID）。 取值范围： 不涉及。
status	String	参数解释： 离线分析任务状态。 取值范围： waiting：任务等待中。 running：任务进行中。 success：任务执行成功。 failed：任务执行失败。
created_at	String	参数解释： 分析任务创建时间。 取值范围： 不涉及。
started_at	String	参数解释： 分析任务开始时间。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间。 取值范围： 不涉及。

状态码：400

表 5-995 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-996 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-997 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-998 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-999 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/offline/key-analysis
```

响应示例

状态码：200

查询离线全量key分析任务列表。

```
{
  "instance_id": "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "total": 1,
  "records": [ {
    "id": "858ee14c-2271-4489-8b82-7bda7459ae3e",
    "status": "success",
    "created_at": "2024-10-15T02:21:18.669Z",
    "started_at": "2024-10-15T02:21:23.534Z",
    "finished_at": "2024-10-15T02:21:25.588Z"
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4230",
  "error_msg": "Does not support offline key analysis."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class ListOfflineKeyAnalysisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListOfflineKeyAnalysisTaskRequest request = new ListOfflineKeyAnalysisTaskRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListOfflineKeyAnalysisTaskResponse response = client.listOfflineKeyAnalysisTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListOfflineKeyAnalysisTaskRequest()
    request.instance_id = "{instance_id}"
    response = client.list_offline_key_analysis_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOfflineKeyAnalysisTaskRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListOfflineKeyAnalysisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询离线全量key分析任务列表。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.17.3 查询离线全量 key 分析详情

功能介绍

查询离线全量key分析详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/instances/{instance_id}/offline/key-analysis/{task_id}

表 5-1000 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 任务ID。可通过DCS控制台进入离线全量key分析界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-1001 响应 Body 参数

参数	参数类型	描述
id	String	参数解释： 任务执行记录ID（此ID对应“查询离线全量key分析详情”参数中的任务ID）。 取值范围： 不涉及。
instance_id	String	参数解释： 实例ID。 取值范围： 不涉及。
node_id	String	参数解释： 节点ID。 取值范围： 不涉及。
backup_id	String	参数解释： 备份ID。 取值范围： 不涉及。
group_name	String	参数解释： 分片名称。 取值范围： 不涉及。
node_ip	String	参数解释： 节点ipv4地址。 取值范围： 不涉及。
node_ipv6	String	参数解释： 节点ipv6地址。 取值范围： 不涉及。
node_type	String	参数解释： 节点类型。 取值范围： master：主节点 slave：从节点

参数	参数类型	描述
analysis_type	String	参数解释： 分析类型。 取值范围： new_backup：新建备份并分析。 old_backup：历史备份文件分析。
started_at	String	参数解释： 分析任务开始时间。 取值范围： 不涉及。
finished_at	String	参数解释： 分析任务结束时间。 取值范围： 不涉及。
largest_key_prefixes	Array of LargestKeyPrefix objects	参数解释： 前缀Key列表。
largest_keys	Array of LargestKey objects	参数解释： 大key列表。
total_bytes	Integer	参数解释： Key的总大小，单位：Bytes。 取值范围： 不涉及。
total_num	Integer	参数解释： Key的总数。 取值范围： 不涉及。
type_bytes	Array of KeyTypeByte objects	参数解释： 每种类型key的总大小，单位：Bytes。 取值范围： 不涉及。
type_num	Array of KeyTypeNum objects	参数解释： 每种类型key总数。 取值范围： 不涉及。

表 5-1002 LargestKeyPrefix

参数	参数类型	描述
key_prefix	String	参数解释： 前缀名称。 取值范围： 不涉及。
type	String	参数解释： Key的数据类型。 取值范围： string list set zset hash
bytes	Integer	参数解释： Key的大小，单位：Bytes。 取值范围： 不涉及。
num	Integer	参数解释： 相同前缀key的数量。 取值范围： 不涉及。

表 5-1003 LargestKey

参数	参数类型	描述
key	String	参数解释： Key名称。 取值范围： 不涉及。

参数	参数类型	描述
type	String	参数解释： Key的数据类型。 取值范围： string list set zset hash
bytes	Integer	参数解释： Key的大小，单位：Bytes。 取值范围： 不涉及。
num_of_elem	Integer	参数解释： 元素数量或元素大小（String类型为元素大小，单位：Bytes。非String类型为元素数量）。 取值范围： 不涉及。

表 5-1004 KeyTypeByte

参数	参数类型	描述
type	String	参数解释： Key的数据类型。 取值范围： string list set zset hash
bytes	Integer	参数解释： 每种数据类型Key的总大小，单位：Bytes。 取值范围： 不涉及。

表 5-1005 KeyTypeEnum

参数	参数类型	描述
type	String	参数解释： Key的数据类型。 取值范围： string list set zset hash
num	Integer	参数解释： Key类型的总数量。 取值范围： 不涉及。

状态码：400

表 5-1006 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-1007 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-1008 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-1009 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-1010 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
GET https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/offline/key-analysis/{task_id}
```

响应示例

状态码：200

查询离线全量key分析详情成功。

```
{
  "id": "858ee14c-2271-4489-8b82-7bda7459ae3e",
  "instance_id": "5f9057b5-c330-4ee2-8138-7e69896eeec3",
  "node_id": "ff80808191c06d650191c163c53d3b40",
  "analysis_type": "new_backup",
  "started_at": "2024-10-15T02:21:23.534Z",
  "finished_at": "2024-10-15T02:23:23.534Z",
  "group_name": "group-0",
  "node_ip": "10.10.0.0",
  "node_ipv6": "2407:c080:11f0:12c:7cc:d251:a5f2:80b6",
  "node_type": "master",
  "largest_key_prefixes": [ {
    "type": "string",
    "key_prefix": "dcs-bigkey",
    "bytes": 300,
    "num": 1
  } ],
  "largest_keys": [ {
    "key": "dcs-bigkey-test",
    "type": "string",
    "bytes": 300,
    "num_of_elem": 1
  } ],
  "total_bytes": 300,
  "total_num": 1,
  "type_bytes": [ {
    "type": "string",
    "bytes": 300
  } ],
  "type_num": [ {
    "type": "string",
    "num": 1
  } ]
}
```

状态码：400

非法请求。

```
{
  "error_code": "DCS.4230",
  "error_msg": "Does not support offline key analysis."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;
```



```
public class ShowOfflineKeyAnalysisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowOfflineKeyAnalysisTaskRequest request = new ShowOfflineKeyAnalysisTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            ShowOfflineKeyAnalysisTaskResponse response = client.showOfflineKeyAnalysisTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ShowOfflineKeyAnalysisTaskRequest()
    request.instance_id = "{instance_id}"
    request.task_id = "{task_id}"
    response = client.show_offline_key_analysis_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowOfflineKeyAnalysisTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowOfflineKeyAnalysisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询离线全量key分析详情成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

5.17.4 删除离线全量 key 分析记录

功能介绍

删除离线全量key分析记录。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/instances/{instance_id}/offline/key-analysis/{task_id}

表 5-1011 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目ID。获取方法请参见 获取项目ID 。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
instance_id	是	String	参数解释： 实例ID。可通过DCS控制台进入实例详情界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。
task_id	是	String	参数解释： 任务ID。可通过DCS控制台进入离线全量key分析界面查看。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

无

响应参数

状态码：200

表 5-1012 响应 Body 参数

参数	参数类型	描述
task_id	String	参数解释： 离线全量key分析任务ID。 取值范围： 不涉及。

状态码：400

表 5-1013 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：401

表 5-1014 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：403

表 5-1015 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：404

表 5-1016 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

状态码：500

表 5-1017 响应 Body 参数

参数	参数类型	描述
error_msg	String	参数解释： 错误信息。 取值范围： 不涉及。
error_code	String	参数解释： 错误码。 取值范围： 请参见 错误码 。
error_ext_msg	String	参数解释： 扩展错误信息（暂未使用，赋值为 null）。 取值范围： 不涉及。

请求示例

```
DELETE https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/offline/key-analysis/{task_id}
```

响应示例

状态码：200

删除离线全量key分析任务成功。

```
{
  "task_id" : "858ee14c-2271-4489-8b82-7bda7459ae3e"
}
```

状态码：400

非法请求。

```
{
  "error_code" : "DCS.4230",
  "error_msg" : "Does not support offline key analysis."
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class DeleteOfflineKeyAnalysisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR_REGION>"))
            .build();
        DeleteOfflineKeyAnalysisTaskRequest request = new DeleteOfflineKeyAnalysisTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            DeleteOfflineKeyAnalysisTaskResponse response = client.deleteOfflineKeyAnalysisTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```



```
client = DcsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DcsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteOfflineKeyAnalysisTaskRequest()
    request.instance_id = "{instance_id}"
    request.task_id = "{task_id}"
    response = client.delete_offline_key_analysis_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := dcs.NewDcsClient(
        dcs.DcsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteOfflineKeyAnalysisTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.DeleteOfflineKeyAnalysisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	删除离线全量key分析任务成功。
400	非法请求。
401	认证信息非法。
403	请求被拒绝访问。
404	所请求的资源不存在。
500	内部服务错误。

错误码

请参见[错误码](#)。

6 权限策略和授权项

如果您需要对您所拥有的DCS进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用DCS服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为角色和策略。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询云服务器列表，那么这个IAM用户被授予的策略中必须包含允许“dcs:servers:list”的授权项，该接口才能调用成功。

支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项

对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。管理员可以在授权项列表中查看授权项是否支持IAM项目或企业项目，“√”表示支持，“×”表示暂不支持。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。

DCS的支持自定义策略授权项如表6-1所示。

表 6-1 DCS 授权项明细

权限	授权项	对应API接口	IA M 项 目 (Pr oje ct)	企 业 项 目 (E nte rpr ise Pro jec t)
创建缓存实例	dcs:instance:create	POST /v2/{project_id}/instances	√	√
查询指定实例	dcs:instance:get	GET /v2/{project_id}/instances/{instance_id}	√	√
修改实例信息	dcs:instance:modify	PUT /v2/{project_id}/instances/{instance_id}	√	√
删除实例	dcs:instance:delete	DELETE /v2/{project_id}/instances/{instance_id}	√	√
缓存实例扩容	dcs:instance:scale	POST /v2/{project_id}/instances/{instance_id}/resize	√	√
查询实例列表	dcs:instance:list	GET /v2/{project_id}/instances	√	√
查询实例配置信息	dcs:instance:getConfiguration	GET /v2/{project_id}/instances/{instance_id}/configs	√	√
修改实例配置信息	dcs:instance:modifyConfiguration	PUT /v2/{project_id}/instances/{instance_id}/configs	√	√
重启实例或清空数据	dcs:instance:modifyStatus	PUT /v2/{project_id}/instances/status	√	√
修改实例密码	dcs:instance:modifyAuthInfo	PUT /v2/{project_id}/instances/{instance_id}/password	√	√

权限	授权项	对应API接口	IA M 项目 (Pr oje ct)	企 业 项目 (E nte rpr ise Pro jec t)
备份实例数据	dc:instance:backupData	POST /v2/{project_id}/instances/{instance_id}/backups	√	√
恢复实例数据	dc:instance:restoreData	POST /v2/{project_id}/instances/{instance_id}/restores	√	√
查询备份任务	dc:instance:getDataBackupLog	GET /v2/{project_id}/instances/{instance_id}/backups	√	√
查询恢复任务	dc:instance:getDataRestoreLog	GET /v2/{project_id}/instances/{instance_id}/restores	√	√
删除备份文件	dc:instance:deleteDataBackupFile	DELETE /v2/{project_id}/instances/{instance_id}/backups/{backup_id}	√	√
查询实例后台任务	dc:instance:getBackgroundTask	GET /v2/{project_id}/instances/{instance_id}/tasks	√	√
删除后台任务	dc:instance:deleteBackgroundTask	DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	√	√
重置实例密码	dc:instance:resetAuthInfo	POST /v2/{project_id}/instances/{instance_id}/password/reset	√	√
下载备份文件	dc:instance:downloadBackupData	POST /v2/{project_id}/instances/{instance_id}/backups/{backup_id}/links	√	√
实例数据迁移	dc:instance:migrateData	-	√	√
Web CLI	dc:instance:webcli	-	√	√
创建迁移任务	dc:migrationTask:create	POST /v2/{project_id}/migration-task	√	X

权限	授权项	对应API接口	IA M 项目 (Pr oje ct)	企 业 项目 (E nte rpr ise Pro jec t)
修改迁移任务配置或停止迁移任务	dc:migrationTask:modify	POST /v2/{project_id}/migration-task/{task_id}/stop	√	X
删除迁移任务	dc:migrationTask:delete	DELETE /v2/{project_id}/migration-tasks/delete	√	X
查询迁移任务列表	dc:migrationTask:list	GET /v2/{project_id}/migration-tasks	√	X
查询迁移任务详情	dc:migrationTask:get	GET /v2/{project_id}/migration-task/{task_id}	√	X
诊断实例	dc:instance:diagnosis	POST /v2/{project_id}/instances/{instance_id}/diagnosis	√	√
查询参数模板列表	dc:template:list	GET /v2/{project_id}/config-templates	√	X
创建参数模板	dc:template:create	POST /v2/{project_id}/config-templates	√	X
实例主备倒换	dc:instance:swap	POST /v2/{project_id}/instances/{instance_id}/swap	√	√
修改实例白名单信息	dc:whitelist:modify	PUT /v2/{project_id}/instance/{instance_id}/whitelist	√	√
获取实例白名单列表	dc:whitelist:list	GET /v2/{project_id}/instance/{instance_id}/whitelist	√	√
创建实例访问账号	dc:aclaccount:create	POST /v2/{project_id}/instances/{instance_id}/accounts	√	√
删除实例访问账号	dc:aclaccount:delete	DELETE /v2/{project_id}/instances/{instance_id}/accounts/{account_id}	√	√

权限	授权项	对应API接口	IA M 项目 (Pr oje ct)	企 业 项目 (E nte rpr ise Pro jec t)
修改实例访问 账号信息	dc:aclaccount:modi fy	PUT /v2/{project_id}/ instances/{instance_id}/ accounts/{account_id}	√	√
获取实例访问 账号列表	dc:aclaccount:list	GET /v2/{project_id}/ instances/{instance_id}/ accounts	√	√
查询慢日志列 表	dc:slowlog:list	GET /v2/{project_id}/ instances/{instance_id}/ slowlog	√	√
查询参数模板	dc:template:get	-	√	X
查询任务进度	dc:job:get	-	√	X
查看实例审计 日志	dc:auditlog:get	-	√	√
查询实例升级 信息	dc:instance:getUpg radeInfo	-	√	√
查询实例SSL信 息	dc:ssl:get	-	√	√
重建域名解析	dc:domainname:re build	-	√	√
实例间交换IP 地址	dc:migrationTask:e xchangelp	-	√	X
实例交换IP后 回滚	dc:migrationTask:r ollbacklp	-	√	X
Kill Redis会话	dc:clients:kill	-	√	√
修改参数模板	dc:template:modify	-	√	X
开启或关闭公 网域名解析	dc:publicdomainna me:update	-	√	√
删除参数模板	dc:template:delete	-	√	X
释放历史域名 解析	dc:histroymainn ame:release	-	√	√

权限	授权项	对应API接口	IA M 项目 (Pr oje ct)	企 业 项目 (E nte rpr ise Pro jec t)
分析实例的大key或者热key	dc:instance:analyze	-	√	√
开启或关闭实例审计日志	dc:auditlog:modify	-	√	√
开启或关闭客户端ip透传	dc:clientiptrans:modify	-	√	√
修改实例SSL信息	dc:ssl:modify	-	√	√
查询Redis会话列表	dc:clients:list	-	√	√
修改实例带宽	dc:bandwidth:modify	-	√	√
查询实例带宽	dc:bandwidth:get	-	√	√
dc: List清单	dc:tag:list	-	√	X
查看实例标签	dc:tag:get	GET /v2/{project_id}/dc/tags	√	√
修改实例标签	dc:tag:modify	-	√	√

7 附录

7.1 状态码

状态码如表7-1所示。

表 7-1 状态码

状态码	编码	错误码说明
100	Continue	继续请求。 这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。
101	Switching Protocols	切换协议。只能切换到更高级的协议。 例如，切换到HTTP的新版本协议。
201	Created	创建类的请求完全成功。
202	Accepted	已经接受请求，但未处理完成。
203	Non-Authoritative Information	非授权信息，请求成功。
204	NoContent	请求完全成功，同时HTTP响应不包含响应体。 在响应OPTIONS方法的HTTP请求时返回此状态码。
205	Reset Content	重置内容，服务器处理成功。
206	Partial Content	服务器成功处理了部分GET请求。
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择。
301	Moved Permanently	永久移动，请求的资源已被永久的移动到新的URI，返回信息会包括新的URI。

状态码	编码	错误码说明
302	Found	资源被临时移动。
303	See Other	查看其它地址。 使用GET和POST请求查看。
304	Not Modified	所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。
305	Use Proxy	所请求的资源必须通过代理访问。
306	Unused	已经被废弃的HTTP状态码。
400	BadRequest	非法请求。 建议直接修改该请求，不要重试该请求。
401	Unauthorized	在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。
402	Payment Required	保留请求。
403	Forbidden	请求被拒绝访问。 返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多的事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。
404	NotFound	所请求的资源不存在。 建议直接修改该请求，不要重试该请求。
405	MethodNotAllowed	请求中带有该资源不支持的方法。 建议直接修改该请求，不要重试该请求。
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。
408	Request Time-out	服务器等候请求时发生超时。 客户端可以随时再次提交该请求而无需进行任何更改。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。
410	Gone	客户端请求的资源已经不存在。 返回该状态码，表明请求的资源已被永久删除。
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息。

状态码	编码	错误码说明
412	Precondition Failed	未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。
414	Request-URI Too Large	请求的URI过长（URI通常为网址），服务器无法处理。
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
416	Requested range not satisfiable	客户端请求的范围无效。
417	Expectation Failed	服务器无法满足Expect的请求头信息。
422	Unprocessable Entity	请求格式正确，但是由于含有语义错误，无法响应。
429	TooManyRequests	表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。
500	InternalServerError	表明服务端能被请求访问到，但是不能理解用户的请求。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	ServiceUnavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。
504	ServerTimeout	请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。
505	HTTP Version not supported	服务器不支持请求的HTTP协议的版本，无法完成处理。

7.2 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

更多服务错误码请参见[API错误中心](#)。

状态码	错误码	错误信息	描述	处理措施
400	DCS.3053	Incorrect node ID.	请使用正确的 Node ID。	请检查Node ID是否正确。
400	DCS.4202	ssl request params are invalid.	SSL接口请求参数错误。	请检查请求参数是否合法。
400	DCS.4933	The param available_zones is invalid.	请求参数AZ非法。	请检查请求参数是否合法。
400	DCS.5050	Shared VPCs are not supported.	不支持共享VPC创建实例。	请检查请求参数是否合法。
400	111400002	Invalid project ID format.	Project ID的格式无效。	请检查项目ID是否合法。
400	111400004	Empty request body.	请求消息体为空。	请检查请求参数必填参数是否填写。
400	111400005	The message body contains invalid characters or is not in JSON format.	请求消息体不是JSON格式或字段非法。	请检查请求参数是否合法。
400	111400007	The selected cache engine type is not supported.	不支持的缓存类型。	请检查请求参数是否合法。
400	111400008	The selected cache engine version is not supported.	不支持的缓存版本。	请检查请求参数是否合法。
400	111400009	Invalid product ID in the request.	请求参数 product_id非法。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400010	Invalid DCS instance name. The name must be 4 to 64 characters in length. Only letters (case-insensitive), digits, underscores (_), and hyphens (-) are allowed.	实例名称不合法，只能包含字母、数字、下划线或者中划线，长度为4-64。	请检查请求参数是否合法。
400	111400011	DCS instance description cannot exceed 1024 characters.	实例描述长度必须为0-1024。	请检查请求参数是否合法。
400	111400012	Invalid capacity parameter in the request.	请求参数 capacity 非法。	请检查请求参数是否合法。
400	111400013	Invalid vpc_id in the request.	请求参数 vpc_id 为空。	请检查请求参数是否合法。
400	111400014	Invalid security_group_id in the request.	请求参数 security_group_id 为空。	请检查请求参数是否合法。
400	111400016	Invalid subnet_id in the request.	请求参数 subnet_id 为空。	请检查请求参数是否合法。
400	111400017	This DCS instance job task is still running.	实例任务还在运行中。	请稍后重试。
400	111400018	This subnet must exist in the VPC.	子网必须在 VPC 中存在。	请检查请求参数是否合法。
400	111400019	The password does not meet complexity requirements.	密码复杂度不符合要求。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400020	DHCP must be enabled for this subnet.	子网的DHCP必须开启。	请检查是否开启DHCP。
400	111400021	The isAutoRenew parameter in the request must be either 0 or 1.	请求参数isAutoRenew非法，只能为0或者1。	请检查请求参数是否合法。
400	111400022	Engine does not match the product id.	缓存引擎与产品ID不匹配。	请检查请求参数是否合法。
400	111400026	This operation is not allowed when the DCS instance is in the current state.	实例的当前状态不支持此操作。	请稍后重试或联系技术支持。
400	111400027	This operation is not supported on this node.	当前节点不支持该操作。	请稍后重试或联系技术支持。
400	111400035	DCS instance quota of the tenant is insufficient.	租户实例配额不足。	请联系技术支持。
400	111400036	Memory quota of the tenant is insufficient.	租户内存配额不足。	请联系技术支持。
400	111400037	The instanceParams parameter in the request contains invalid characters or is not in JSON format.	请求参数instanceParams非法，不是JSON格式或字段非法。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400038	The periodNum parameter in the request must be an integer.	请求参数 periodNum 非法，必须为整数。	请检查请求参数是否合法。
400	111400039	The quota limit has been reached.	请求调整配额超出限制范围。	请检查请求参数是否合法。
400	111400042	This AZ does not exist.	可用区不存在。	请检查请求参数是否合法。
400	111400045	This DCS instance is already unfrozen.	实例没有被冻结，不能进行解除冻结操作。	请检查请求参数是否合法。
400	111400046	This security group does not exist.	安全组不存在。	请检查请求参数是否合法。
400	111400047	The periodType parameter in the request must be either 2 or 3.	请求参数 periodType 非法，必须为2或者3。	请检查请求参数是否合法。
400	111400048	The security group must have both outbound and inbound rules with protocols set to ANY.	安全组规则不符合要求，请确保安全组规则中同时包含协议为“ANY”的出方向和入方向规则。	请检查请求参数是否合法。
400	111400053	the upgrade instance version equals to current version.	请求参数 targetVersion 非法，不能与当前版本相同。	请检查请求参数是否合法。
400	111400054	the selected available zone quota not enough.	当前选择的可用区 DCS 服务已资源不足。	请前往其他可用区选择或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	111400060	This DCS instance name already exists.	实例名称已经存在。	请检查请求参数是否合法。
400	111400061	Invalid instance ID format.	实例ID的格式无效。	请检查请求参数是否合法。
400	111400062	Invalid {0} parameter in the request.	查询参数{0}非法。	请检查请求参数是否合法。
400	111400063	Invalid {0} parameter in the request.	请求参数{0}非法。	请检查请求参数是否合法。
400	111400064	The action parameter in the request must be start, stop, or restart.	请求参数 action非法，只能为start、stop或restart。	请检查请求参数是否合法。
400	111400065	The instances parameter in the request cannot be a null value or left unspecified.	请求参数 instances为空。	请检查请求参数是否合法。
400	111400066	Invalid value of the configure parameter {0}.	配置参数{0}非法。	请检查请求参数是否合法。
400	111400067	The available_zones parameter in the request must an array that contains only one AZ ID.	请求参数 available_zones非法，必须为只包含一个可用区 ID的数组。	请检查请求参数是否合法。
400	111400068	This VPC does not exist.	VPC不存在。	请检查请求参数是否合法。
400	111400070	Invalid task ID format.	任务ID的格式无效。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400072	Invalid saveDays instance backup policy parameter in the request.	请求的实例备份策略参数 saveDays 非法，范围在1-7天。	请检查请求参数是否合法。
400	111400073	Invalid backupType instance backup policy parameter in the request.	请求的实例备份策略参数 backupType 非法，只支持 auto、manual。	请检查请求参数是否合法。
400	111400074	Invalid periodType instance backup policy parameter in the request.	请求的实例备份策略参数 periodType 非法，只支持 weekly。	请检查请求参数是否合法。
400	111400075	Invalid backupAt instance backup policy parameter in the request.	请求的实例备份策略参数 backupAt 非法，不能为空。	请检查请求参数是否合法。
400	111400076	Invalid beginAt instance backup policy parameter in the request.	请求的实例备份策略参数 beginAt 非法，只能是 00:00-00:00 格式。	请检查请求参数是否合法。
400	111400080	Instance password verify failed.	实例密码校验错误。	请检查请求参数是否合法。
400	111400086	only master standby instance is support this action	只有主备类型的实例才支持这个操作。	不支持此操作。
400	111400087	the backup record status is err when restore this instance	恢复实例时，备份记录的状态错误，succeed 状态的才能进行恢复。	请稍后重试或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	111400094	the system is not support dcs job.	系统不支持后台任务功能。	不支持此操作。
400	111400095	the system is not support backup and restore.	系统不支持备份恢复功能。	不支持此操作。
400	111400096	the instance is backuping.	实例正在备份中，请稍后重试。	请稍后重试。
400	111400097	the instance is restoring.	实例正在恢复中，请稍后重试。	请稍后重试。
400	111400098	The value of the remark parameter must be 0 to 128 characters in length.	请求的实例备份策略参数 remark非法，长度0~128。	请检查请求参数是否合法。
400	111400099	DCS instances in the Creating, Starting, Stopping, or Restarting state cannot be deleted.	实例状态为创建中、重启中、删除中时不允许执行删除操作。	请稍后重试。
400	111400100	The number of instance IDs in an instances array cannot exceed 50.	instances数组最多只能包含50个实例ID。	请检查请求参数是否合法。
400	111400102	The system is not support resize.	系统不支持扩容。	不支持此操作。
400	111400103	The system is not support resize.	扩容容量必须比原容量大。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400104	The DCS instance is recovering from an internal fault. Please try again later or contact customer service.	实例内部异常正在恢复，请稍后重试或联系客服。	请稍后重试或联系技术支持。
400	111400105	The value of reserved-memory cannot be greater than the free memory size of this DCS instance.	配置参数 reserved-memory 的值不能大于实例的空闲内存。	请检查请求参数是否合法。
400	111400106	The value of maintain time illegal.	维护时间窗参数非法。	请检查请求参数是否合法。
400	111400107	The instance exists for unpaid scale up orders. Please process non payment orders first.	实例已存在未支付的扩容订单，请先处理未支付订单。	请先处理未支付订单。
400	111400108	The Instance exists for processing scale up order. Please try again later.	实例存在正在处理的扩容订单，请稍后重试。	请稍后重试或联系技术支持。
400	111400111	the instance is restarting.	实例正在重启中，请稍后重试。	请稍后重试或联系技术支持。
400	111400113	the instance is extending.	实例正在扩容中，请稍后重试。	请稍后重试或联系技术支持。
400	111400114	the instance is configuring.	正在修改实例配置，请稍后重试。	请稍后重试或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	111400115	the instance is changing the password.	正在修改实例密码，请稍后重试。	请稍后重试或联系技术支持。
400	111400116	the instance is upgrading.	实例正在升级中，请稍后重试。	请稍后重试或联系技术支持。
400	111400117	the instance is rollbacking the version.	实例正在回滚中，请稍后重试。	请稍后重试或联系技术支持。
400	111400118	the instance is creating.	实例正在创建中，请稍后重试。	请稍后重试或联系技术支持。
400	111400119	Query Bill Sample failed	实例不存在。	请检查请求参数是否合法。
400	111400120	the instance is Freezing.	实例正在冻结中，请稍后重试。	请稍后重试或联系技术支持。
400	111400800	Invalid {0} in the request.	请求参数{0}非法。	请检查请求参数是否合法。
400	111400843	The no_password_access parameter is missing or its value is invalid.	缺少参数 no_password_access或取值不合法。	请检查请求参数是否合法。
400	111400844	The access_user parameter is missing or its value is invalid.	缺少参数 access_user或取值不合法。	请检查请求参数是否合法。
400	111400845	The password parameter is missing or its value is invalid.	缺少参数 password或取值不合法。	请检查请求参数是否合法。
400	111400849	The request parameter new_password should not exist.	请求参数 new_password不应存在。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	111400850	This operation is not supported when Password-Free Access is enabled for the instance.	实例当前的免密访问设置不支持该操作。	请使用重置密码。
400	111400857	DCS instances cannot be deleted if they are billed on a yearly/monthly basis.	包年/包月实例不支持删除。	请使用退订功能删除包年/包月实例
400	DCS. 4855	Master standby swap is not supported.	不支持主备切换。	不支持此操作。
400	DCS.4002	Invalid project ID format.	Project ID的格式无效。	请检查请求参数是否合法。
400	DCS.4004	Empty request body.	请求消息体为空。	请检查请求参数是否合法。
400	DCS.4005	The message body contains invalid characters or is not in JSON format.	请求消息体不是JSON格式或字段非法。	请检查请求参数是否合法。
400	DCS.4006	The selected cache engine type is not supported.	不支持的缓存引擎。	请检查请求参数是否合法。
400	DCS.4007	Instance type is not supported.	不支持的缓存类型。	请检查请求参数是否合法。
400	DCS.4008	The selected cache engine version is not supported.	不支持的缓存版本。	请检查请求参数是否合法。
400	DCS.4009	Invalid product ID in the request.	请求参数 product_id非法。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4010	Invalid DCS instance name. The name must be 4 to 64 characters in length. Only letters (case-insensitive), digits, underscores (_), and hyphens (-) are allowed.	实例名称不合法，只能包含字母、数字、下划线或者中划线，长度为4-64。	请检查请求参数是否合法。
400	DCS.4011	DCS instance description cannot exceed 1024 characters.	实例描述长度必须为0-1024。	请检查请求参数是否合法。
400	DCS.4012	Invalid capacity parameter in the request.	请求参数 capacity 非法。	请检查请求参数是否合法。
400	DCS.4013	Invalid vpc_id in the request.	请求参数 vpc_id 为空。	请检查请求参数是否合法。
400	DCS.4014	Invalid security_group_id in the request.	请求参数 security_group_id 为空。	请检查请求参数是否合法。
400	DCS.4016	Invalid subnet_id in the request.	请求参数 subnet_id 为空。	请检查请求参数是否合法。
400	DCS.4017	This DCS instance job task is still running.	实例任务还在运行中。	请稍后重试。
400	DCS.4018	This subnet must exist in the VPC.	子网必须在 VPC 中存在。	请检查请求参数是否合法。
400	DCS.4019	The password does not meet complexity requirements.	密码复杂度不符合要求。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4020	DHCP must be enabled for this subnet.	子网的DHCP必须开启。	请检查DHCP是否开启。
400	DCS.4021	The isAutoRenew parameter in the request must be either 0 or 1.	请求参数 isAutoRenew 非法，只能为0或者1。	请检查请求参数是否合法。
400	DCS.4022	Engine does not match the product id.	缓存引擎与产品ID不匹配。	请检查请求参数是否合法。
400	DCS.4025	The end time must be after the start time.	结束时间不能早于开始时间。	请检查请求参数是否合法。
400	DCS.4026	This operation is not allowed when the DCS instance is in the current state.	实例的当前状态不支持此操作。	请稍后重试或联系技术支持。
400	DCS.4027	This operation is not supported on this node.	当前节点不支持该操作。	请稍后重试或联系技术支持。
400	DCS.4028	The diagnosis time range is max 10 minutes.	诊断时间范围应在10分钟内。	请检查请求参数是否合法。
400	DCS.4030	The start time must be within the last seven days.	开始时间不能早于7天前。	请检查请求参数是否合法。
400	DCS.4031	The start time cannot be later than the current server time.	开始时间不能晚于服务器当前时间。	请检查请求参数是否合法。
400	DCS.4033	The start time must be after the instance start time.	开始时间不能早于实例启动时间。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4035	DCS instance quota of the tenant is insufficient.	租户实例配额不足。	请联系技术支持。
400	DCS.4036	Memory quota of the tenant is insufficient.	租户内存配额不足。	请联系技术支持。
400	DCS.4037	The instanceParams parameter in the request contains invalid characters or is not in JSON format.	请求参数 instanceParams 非法，不是 JSON 格式或字段非法。	请检查请求参数是否合法。
400	DCS.4038	The periodNum parameter in the request must be an integer.	请求参数 periodNum 非法，必须为整数。	请检查请求参数是否合法。
400	DCS.4039	The quota limit has been reached.	请求调整配额超出限制范围。	请检查请求参数是否合法。
400	DCS.4042	This AZ does not exist.	可用区不存在。	请检查请求参数是否合法。
400	DCS.4045	This DCS instance is already unfrozen.	实例没有被冻结，不能进行解除冻结操作。	请稍后重试或联系技术支持。
400	DCS.4046	This security group does not exist.	安全组不存在。	请检查请求参数是否合法。
400	DCS.4047	The periodType parameter in the request must be either 2 or 3.	请求参数 periodType 非法，必须为 2 或者 3。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4048	The security group must have both outbound and inbound rules with protocols set to ANY.	安全组规则不符合要求，请确保安全组规则中同时包含协议为“ANY”的出方向和入方向规则。	请检查请求参数是否合法。
400	DCS.4049	The instance status is not running.	实例状态不是运行中。	请联系技术支持。
400	DCS.4053	the upgrade instance version equals to current version.	请求参数targetVersion非法，不能与当前版本相同。	请检查请求参数是否合法。
400	DCS.4054	the selected available zone quota not enough.	当前选择的可用区DCS服务已售罄。	请前往其他可用区选择或联系技术支持。
400	DCS.4060	This DCS instance name already exists.	实例名称已经存在。	请检查请求参数是否合法。
400	DCS.4061	Invalid instance ID format.	实例ID格式非法。	请检查请求参数是否合法。
400	DCS.4062	Invalid {0} parameter in the request.	查询参数{0}非法。	请检查请求参数是否合法。
400	DCS.4063	Invalid {0} parameter in the request.	请求参数{0}非法。	请检查请求参数是否合法。
400	DCS.4064	The action parameter in the request must be start, stop, or restart.	请求参数action非法，只能为start、stop或restart。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4065	The instances parameter in the request cannot be a null value or left unspecified.	请求参数 instances 为空。	请检查请求参数是否合法。
400	DCS.4066	Invalid value of the configure parameter {0}.	配置参数{0}非法。	请检查请求参数是否合法。
400	DCS.4067	The available_zones parameter in the request must an array that contains only one AZ ID.	请求参数 available_zones 非法，必须为只包含一个可用区 ID 的数组。	请检查请求参数是否合法。
400	DCS.4068	This VPC does not exist.	VPC 不存在。	请检查请求参数是否合法。
400	DCS.4070	Invalid task ID format.	任务 ID 的格式无效。	请检查请求参数是否合法。
400	DCS.4072	Invalid saveDays instance backup policy parameter in the request.	请求的实例备份策略参数 saveDays 非法，范围在 1-7 天。	请检查请求参数是否合法。
400	DCS.4073	Invalid backupType instance backup policy parameter in the request.	请求的实例备份策略参数 backupType 非法，只支持 auto、manual。	请检查请求参数是否合法。
400	DCS.4074	Invalid periodType instance backup policy parameter in the request.	请求的实例备份策略参数 periodType 非法，只支持 weekly。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4075	Invalid backupAt instance backup policy parameter in the request.	请求的实例备份策略参数 backupAt 非法，不能为空。	请检查请求参数是否合法。
400	DCS.4076	Invalid beginAt instance backup policy parameter in the request.	请求的实例备份策略参数 beginAt 非法，只能是 00:00-00:00 格式。	请检查请求参数是否合法。
400	DCS.4080	Instance password verify failed.	实例密码校验错误。	请检查请求参数是否合法。
400	DCS.4086	only master standby instance is support this action	只有主备类型的实例才支持这个操作。	不支持此操作。
400	DCS.4087	the backup record status is err when restore this instance	恢复实例时，备份记录的状态错误，succeed 状态的才能进行恢复。	请稍后重试或联系技术支持。
400	DCS.4094	the system is not support dcs job.	系统不支持后台任务功能。	不支持此操作。
400	DCS.4095	the system is not support backup and restore.	系统不支持备份恢复功能。	不支持此操作。
400	DCS.4096	the instance is backuping.	实例正在备份中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4097	the instance is restoring.	实例正在恢复中，请稍后重试。	请稍后重试或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4098	The value of the remark parameter must be 0 to 128 characters in length.	请求的实例备份策略参数remark非法，长度0~128。	请检查请求参数是否合法。
400	DCS.4099	DCS instances in the Creating, Starting, Stopping, or Restarting state cannot be deleted.	实例状态为创建中、重启中、删除中时不允许执行删除操作。	请稍后重试。
400	DCS.4100	The number of instance IDs in an instances array cannot exceed 50.	instances数组最多只能包含50个实例ID。	请检查请求参数是否合法。
400	DCS.4102	The system is not support resize.	系统不支持扩容。	不支持此操作。
400	DCS.4103	The system is not support resize.	扩容容量必须比原容量大。	请检查请求参数是否合法。
400	DCS.4104	The DCS instance is recovering from an internal fault. Please try again later or contact customer service.	实例内部异常正在恢复，请稍后重试或联系技术支持。	请稍后重试或联系技术支持。
400	DCS.4105	The value of reserved-memory cannot be greater than the free memory size of this DCS instance.	配置参数reserved-memory的值不能大于实例的空闲内存。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4106	The value of maintain time illegal.	维护时间窗参数非法。	请检查请求参数是否合法。
400	DCS.4107	The instance exists for unpaid scale up orders. Please process non payment orders first.	实例已存在未支付的扩容订单，请先处理未支付订单。	请先处理未支付订单。
400	DCS.4108	The Instance exists for processing scale up order. Please try again later.	实例存在正在处理的扩容订单，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4111	the instance is restarting.	实例正在重启中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4113	the instance is extending.	实例正在扩容中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4114	the instance is configuring.	正在修改实例配置，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4115	the instance is changing the password.	正在修改实例密码，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4116	the instance is upgrading.	实例正在升级中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4117	the instance is rollbacking the version.	实例正在回滚中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4118	the instance is creating.	实例正在创建中，请稍后重试。	请稍后重试或联系技术支持。
400	DCS.4119	Query Bill Sample failed	实例不存在。	请检查请求参数是否合法。
400	DCS.4120	the instance is Freezing.	实例正在冻结中，请稍后重试。	请稍后重试或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4200	Instance type cannot be changed.	当前实例不支持变更实例类型。	请检查请求参数是否合法。
400	DCS.4253	All replicas have a priority of 0.	实例从节点主备切换优先级都为0，禁止主备切换。	请检查请求参数是否合法。
400	DCS.4300	Client IP transparent transmission is being processed, please do not submit it repeatedly.	客户端IP正在处理，请勿重复提交。	客户端IP透传正在开启或关闭中。
400	DCS.4800	Invalid {0} in the request.	请求参数{0}非法。	请检查请求参数是否合法。
400	DCS.4819	The first_scan_at parameter is invalid. Its must be ISO format utc date-time. The time must be later than the current time.	参数first_scan_at取值不合法，不符合ISO格式的UTC时间，first_scan_at必须晚于当前时间。	请检查请求参数是否合法。ISO UTC格式示例 "2021-08-18T07:33:00.000Z"。
400	DCS.4820	The interval parameter is invalid. must Greater than or equal to 0.	参数interval取值不合法。必须大于等于0秒。	请检查请求参数是否合法。
400	DCS.4821	The timeout parameter is invalid. Its must Greater than or equal to interval * 2, if interval is 0, timeout must Greater than 1 second.	参数timeout取值不合法，必须大于0，且至少是扫描间隔时间的两倍。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4822	The scan_keys_count parameter is invalid. Its must Greater than or equal to 10, and Less than or equal to 1000.	参数 scan_keys_count取值不合法，取值范围 10~1000。	请检查参数请求是否合法。
400	DCS.4838	Do not support feature client ip transparent transmission.	不支持客户端 IP透传特性。	客户端IP透传特性开关未开。
400	DCS.4839	Do not support feature public dns.	不支持公网域名解析特性。	公网域名解析特性开关未开。
400	DCS.4840	Do not support feature dns.	不支持域名解析特性。	域名解析特性开关未开。
400	DCS.4841	Instance do not support client ip transparent transmission.	实例不支持客户端IP透传。	实例版本较低不支持客户端IP透传。
400	DCS.4843	The no_password_access parameter is missing or its value is invalid.	缺少参数 no_password_access或取值不合法。	请检查请求参数是否合法。
400	DCS.4844	The access_user parameter is missing or its value is invalid.	缺少参数 access_user或取值不合法。	请检查请求参数是否合法。
400	DCS.4845	The password parameter is missing or its value is invalid.	缺少参数 password或取值不合法。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4846	Client ip transparent transmission is already closed,please do not repeat the operation.	客户端IP透传已关闭，请勿重复操作。	客户端IP透传当前已关闭。
400	DCS.4847	Client ip transparent transmission is already opened,please do not repeat the operation.	客户端IP透传已开启，请勿重复操作。	客户端IP透传当前已开启。
400	DCS.4848	Client Management is not supported.	不支持会话管理功能。	不支持此操作。
400	DCS.4849	The request parameter new_password should not exist.	请求参数 new_password 不应存在。	请检查请求参数是否合法。
400	DCS.4850	This operation is not supported when Password-Free Access is enabled for the instance.	实例当前的免密访问设置不支持该操作。	请检查请求参数是否合法。
400	DCS.4857	DCS instances cannot be deleted if they are billed on a yearly/monthly basis.	包年/包月实例不支持删除。	请检查请求参数是否合法。
400	DCS.4875	Create replication number exceed max number limit.	创建副本数超过最大数量限制。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4879	r/w instance have only one repl can not remove ip from dns.	读写分离实例只有一个副本不支持摘除副本IP。	请检查请求参数是否合法。
400	DCS.4911	The network ip insufficient.	子网ip数量不足。	请检查子网ip数量是否充足，建议释放不再使用的ip或者更换ip数量充足的子网
400	DCS.4918	Instance bigkey analyze is running.	实例大key分析正在执行。	请稍后重试。
400	DCS.4919	Does not support bigkey analyze.	不支持大key分析。	不支持此操作。
400	DCS.4927	The param spec_code is invalid	请求参数spec_code非法。	请检查请求参数是否合法。
400	DCS.4930	The request param node_list is invalid	请求参数node_list非法。	请检查请求参数是否合法。
400	DCS.4931	Node is not replica, can't delete.	节点非只读副本，无法删除。	请检查请求参数是否合法。
400	DCS.4935	Only one slave replication has dns ip, can not delete	只有一个备节点有dns ip，不允许删除该副本。	请检查请求参数是否合法。
400	DCS.4936	The master node cannot be deleted	主节点不允许被删除。	请检查请求参数是否合法。
400	DCS.4937	Only has one slave node,can not delete	唯一的备节点不允许被删除。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
400	DCS.4939	The param slave_priority_weight is invalid.	请求参数 slave_priority_weight非法。	请检查请求参数是否合法。
400	DCS.4941	The hotkey id does not exist.	热key分析ID不存在。	请检查请求参数是否合法。
400	DCS.4942	The bigkey id does not exist.	大key分析ID不存在。	请检查请求参数是否合法。
400	DCS.4957	This operation is not allowed when the task is in the current state.	缓存分析任务的当前状态不支持此操作。	只有成功状态的缓存分析任务支持此操作。
400	DCS.4998	Domain names are up to date.	域名已是最新。	域名已最新，不需要重建。
400	DCS.4999	Do not support feature update domain name.	不支持域名优化特性。	域名优化特性开关未开。
401	111401001	Invalid token.	Token无效。	请检查请求参数是否合法。
401	111401002	Token expired.	Token过期。	请检查请求参数是否合法。
401	111401003	No token in the request.	Token缺失。	请检查请求参数是否合法。
401	111401004	Project ID does not match the token.	Project-ID和Token不匹配。	请检查请求参数是否合法。
401	DCS.1001	Invalid token.	Token无效。	请检查请求参数是否合法。
401	DCS.1002	Token expired.	Token过期。	请检查请求参数是否合法。
401	DCS.1003	No token in the request.	Token缺失。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
401	DCS.1004	Project ID does not match the token.	Project-ID和Token不匹配。	请检查请求参数是否合法。
403	111403002	This tenant has read permission only and cannot perform this operation.	租户只有只读权限，无法操作DCS。	请检查请求参数是否合法。
403	111403003	This role does not have the permission to perform this operation.	角色没有操作权限，无法执行此操作。	请检查请求参数是否合法。
403	DCS.2003	This role does not have the permission to perform this operation.	角色没有操作权限，无法执行此操作。	请检查请求参数是否合法。
403	DCS.2007	Policy does not allow {} to be performed.	操作权限不足。	请检查账户是否有对应的操作权限。
403	DCS.3002	This tenant has read permission only and cannot perform this operation.	租户只有只读权限，无法操作DCS。	请检查请求参数是否合法。
403	DCS.3003	This role does not have the permission to perform this operation.	角色没有操作权限，无法执行此操作。	请检查请求参数是否合法。
404	111404001	The requested URL does not exist.	请求的URL不存在。	请检查请求参数是否合法。
404	111404022	This DCS instance does not exist.	实例不存在。	请检查请求参数是否合法。

状态码	错误码	错误信息	描述	处理措施
404	111404023	This DCS order does not exist.	订单不存在。	请检查请求参数是否合法。
404	DCS.3022	This DCS instance does not exist.	实例不存在。	请检查请求参数是否合法。
404	DCS.4001	The requested URL does not exist.	请求的URL不存在。	请检查请求参数是否合法。
404	DCS.4023	This DCS order does not exist.	订单不存在。	请检查请求参数是否合法。
404	DCS.5101	Create agency failed.	创建委托授权失败。	请稍后重试或联系技术支持。
405	111405001	This request method is not allowed.	请求中指定的方法不被允许。	请检查请求参数是否合法。
405	DCS.5001	This request method is not allowed.	请求中指定的方法不被允许。	请检查请求参数是否合法。
500	111400069	Another user is modifying configuration parameters of the DCS instance. Please try again later.	其他用户正在修改实例的配置参数，请稍后再试。	请稍后重试。
500	111400101	Delete instance backup file failed.	删除实例备份文件失败。	请联系技术支持。
500	111400842	job execution status failed.	Job任务执行失败。	请联系技术支持。
500	111500000	Internal service error.	内部服务错误。	请联系技术支持。
500	111500006	Internal service error.	内部服务错误。	请联系技术支持。
500	111500017	Internal service error.	内部服务错误。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	111500020	vm add port fail	虚拟机添加端口异常。	请联系技术支持。
500	111500024	Internal service error.	内部服务错误。	请联系技术支持。
500	111500031	create instance fail	创建DCS实例失败。	请联系技术支持。
500	111500032	Internal service error.	内部服务错误。	请联系技术支持。
500	111500037	query order info fail	查询订单详情失败。	请稍后重试或联系技术支持。
500	111500041	No resource tenant available.	无可用资源租户。	请稍后重试或联系技术支持。
500	111500044	update resource status fail	更新云服务实例资源状态失败。	请稍后重试或联系技术支持。
500	111500053	Internal service error.	内部服务错误。	请联系技术支持。
500	111500054	Internal service error.	内部服务错误。	请联系技术支持。
500	111500070	Internal service error.	内部服务错误。	请联系技术支持。
500	111500071	Internal service error.	内部服务错误。	请联系技术支持。
500	111500077	Internal service error.	内部服务错误。	请联系技术支持。
500	111500078	Internal service error.	内部服务错误。	请联系技术支持。
500	111500079	Internal service error.	内部服务错误。	请联系技术支持。
500	111500082	Internal service error.	内部服务错误。	请联系技术支持。
500	111500083	Internal service error.	内部服务错误。	请联系技术支持。
500	111500085	Internal service error.	内部服务错误。	请联系技术支持。
500	111500090	Internal service error.	内部服务错误。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	111500091	Internal service error.	内部服务错误。	请联系技术支持。
500	111500092	Internal service error.	内部服务错误。	请联系技术支持。
500	111500095	Internal service error.	内部服务错误。	请联系技术支持。
500	111500104	Internal service error.	内部服务错误。	请联系技术支持。
500	DCS. 5094	Master standby swap failed.	主备切换失败。	请联系技术支持。
500	DCS.4069	Another user is modifying configuration parameters of the DCS instance. Please try again later.	其他用户正在修改实例的配置参数，请稍后再试。	请稍后重试。
500	DCS.4101	Delete instance backup file failed.	删除实例备份文件失败。	请联系技术支持。
500	DCS.4842	job execution status failed.	Job任务执行失败。	请联系技术支持。
500	DCS.5000	Internal service error.	内部服务错误。	请联系技术支持。
500	DCS.5006	Submit job failed.	提交任务失败。	请联系技术支持。
500	DCS.5017	Failed to save instance information.	保存实例信息失败	请联系技术支持。
500	DCS.5020	vm add port fail	虚拟机添加端口异常。	请联系技术支持。
500	DCS.5024	Query instance failed.	查询实例失败。	请联系技术支持。
500	DCS.5031	create instance fail	创建DCS实例失败。	请联系技术支持。
500	DCS.5032	Failed to create order.	创建订单号失败。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	DCS.5037	query order info fail	查询订单详情失败。	请联系技术支持。
500	DCS.5039	Obtain IAM service failed.	获取IAM服务失败。	请联系技术支持。
500	DCS.5040	Call IAM API for querying agency failed.	调用IAM查询委托接口失败。	请联系技术支持。
500	DCS.5041	No resource tenant available.	无可用资源租户。	请联系技术支持。
500	DCS.5042	Call IAM API for creating agency failed.	调用IAM创建委托接口失败。	请联系技术支持。
500	DCS.5043	Call IAM API for querying permission failed.	调用IAM查询权限接口失败。	请联系技术支持。
500	DCS.5044	update resource status fail	更新云服务实例资源状态失败。	请联系技术支持。
500	DCS.5045	Call IAM API for adding project-level agency permissions failed.	调用IAM增加项目级别的委托权限失败。	请联系技术支持。
500	DCS.5053	Instance node not found.	未找到实例节点。	请联系技术支持。
500	DCS.5054	Failed to generate password.	生成密码失败。	请联系技术支持。
500	DCS.5070	Internal service error.	内部服务错误。	请联系技术支持。
500	DCS.5071	Failed to create instance backup strategy.	创建实例备份策略失败。	请联系技术支持。
500	DCS.5077	Query instance failed.	查询实例备份策略失败。	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	DCS.5078	Query backup record failed.	查询备份记录失败。	请联系技术支持。
500	DCS.5079	Query group node failed.	查询节点错误。	请联系技术支持。
500	DCS.5081	The resource tenant does not exist.	资源租户不存在。	请联系技术支持。
500	DCS.5082	Failed to register resource tenant backup user.	注册资源租户备份用户失败。	请联系技术支持。
500	DCS.5083	Failed to save backup user.	保存备份用户失败。	请联系技术支持。
500	DCS.5085	Failed to update backup strategy.	更新备份策略失败。	请联系技术支持。
500	DCS.5090	Failed to save backup information.	保存备份信息失败。	请联系技术支持。
500	DCS.5091	Script execution failed.	执行脚本失败。	请联系技术支持。
500	DCS.5092	Failed to save and restore data.	保存恢复数据失败。	请联系技术支持。
500	DCS.5095	The script execution failed and needs to be retried.	执行脚本失败，需要重试。	请联系技术支持。
500	DCS.5104	Failed to modify order.	修改订单失败。	请联系技术支持。

7.3 获取项目 ID

您可以通过[调用API获取项目ID](#)或[从控制台获取项目ID](#)两种方式获取项目ID。

调用 API 获取项目 ID

项目ID可以通过调用[查询指定条件下的项目信息](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目ID。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "XXXXX",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

从控制台获取项目 ID

在调用接口的时候，部分URL中需要填入项目ID（project_id），所以需要先在管理控制台上获取到项目ID。

项目ID获取步骤如下：

- 步骤1** 注册并登录管理控制台。
- 步骤2** 鼠标悬停在右上角的用户名，在下拉列表中单击“我的凭证”。
- 步骤3** 在“API凭证”页面的项目列表中查看项目ID。

图 7-1 查看项目 ID



----结束

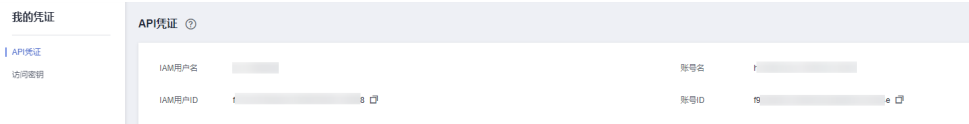
7.4 获取账户名和账号 ID

在调用接口的时候，部分URL中需要填入账户名和账号ID，所以需要先在管理控制台上获取到账户名和账号ID。账户名和账号ID获取步骤如下：

- 1. 注册并登录管理控制台。

2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。
在“API凭证”页面查看账号名和账号ID。

图 7-2 查看账号名和账号 ID



7.5 缓存实例状态说明

表 7-2 缓存实例状态说明

状态	说明
CREATING	申请缓存实例后，在缓存实例状态进入运行中之前的状态。
CREATEFAILED	缓存实例处于创建失败的状态。
RUNNING	缓存实例正常运行状态。 在这个状态的实例可以运行您的业务。
ERROR	缓存实例处于故障的状态。
RESTARTING	缓存实例正在进行重启操作。
FROZEN	缓存实例处于已冻结状态，用户可以在“我的订单”中续费 开启冻结的缓存实例。
EXTENDING	缓存实例处于正在扩容的状态。
RESTORING	缓存实例数据恢复中的状态。
FLUSHING	缓存实例数据清空中的状态。

7.6 实例诊断 ID 说明

表 7-3 cause_ids

ID	ID类型	ID说明
1	cause	活跃连接数超过{THRESHOLD}
2	cause	瞬时流量过大
3	cause	底层磁盘异常
4	cause	持久化失败
5	cause	内存使用量超过{THRESHOLD}
6	cause	已开启RDB,内存使用量超过{THRESHOLD}

ID	ID类型	ID说明
7	cause	Redis内部内存碎片率过高
8	cause	缓存命中率低于{THRESHOLD}
9	cause	大量key集中过期
10	cause	执行了O(N)时间复杂度命令：{COMMANDS}
11	cause	CPU负载高于{THRESHOLD}
12	cause	执行持久化
13	cause	QPS升高

表 7-4 impact_ids

ID	ID类型	ID说明
1	impact	Redis拒绝连接
2	impact	Redis断开连接
3	impact	Redis响应变慢
4	impact	RDB持久化失败
5	impact	缓存命中率下降
6	impact	AOF持久化失败

表 7-5 advice_ids

ID	ID类型	ID说明
1	advice	客户端使用连接池复用已有连接
2	advice	采用读写分离或集群
3	advice	扩容
4	advice	业务低峰期手动执行memory purge
5	advice	排查Redis使用方法或降低缓存粒度，减少内存逐出
6	advice	分散设置key过期时间
7	advice	避免使用O(N)时间复杂度命令

8 历史 API

8.1 生命周期管理

8.1.1 创建缓存实例

功能介绍

创建缓存实例，该接口创建的缓存实例支持按需计费和包周期两种方式。

URI

POST /v1.0/{project_id}/instances

参数说明见[表8-1](#)。

表 8-1 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。获取方式，参考 获取项目ID 。

请求消息

请求参数

参数说明见[表8-2](#)。

表 8-2 参数说明

参数	类型	必选	说明
name	String	是	实例名称。 由英文字符开头，只能由英文字母、数字、中划线和下划线组成。 创建单个实例时，名称长度为4到64位的字符串。批量创建实例时，名称长度为4到56位的字符串，且实例名称格式为“自定义名称- <i>n</i> ”，其中 <i>n</i> 从000开始，依次递增。例如，批量创建两个实例，自定义名称为dcs_demo，则两个实例的名称为dcs_demo-000和dcs_demo-001。
description	String	否	实例的描述信息。 长度不超过1024的字符串。 说明 \"与\"在json报文中属于特殊字符，如果参数值中需要显示\"或者\"字符，请在字符前增加转义字符\\，比如\\\"或者\\\"。
engine	String	是	缓存引擎：Redis和Memcached。
engine_version	String	是	缓存版本。当缓存引擎为Redis时，该字段为必填，取值为3.0、4.0或5.0。当缓存引擎为Memcached时，该字段为可选，取值可为空。
capacity	Float	是	缓存容量（G Byte）。 - Redis3.0：单机和主备类型实例取值：2、4、8、16、32、64。Proxy集群实例规格支持64、128、256、512和1024。 - Redis4.0和Redis5.0：单机和主备类型实例取值：0.125、0.25、0.5、1、2、4、8、16、32、64。Cluster集群实例规格支持24、32、48、64、96、128、192、256、384、512、768、1024。 - Memcached：单机和主备类型实例取值：2、4、8、16、32、64。
no_password_access	String	是	是否允许免密码访问缓存实例。 - true：该实例无需密码即可访问。 - false：该实例必须通过密码认证才能访问。 若未配置该参数则默认值为“false”。

参数	类型	必选	说明
password	String	否	缓存实例的认证信息。 说明 当“no_password_access”配置为“false”或未配置时，请求消息中须包含password参数。 Redis类型的缓存实例密码复杂度要求： • 输入长度为8到32位的字符串。 • 新密码不能与旧密码相同。 • 必须包含如下四种字符中的三种组合： - 小写字母 - 大写字母 - 数字 - 特殊字符包括（`~!@#\$\$%^&*()-_+=\ [{}`:;',<.>/?）
access_user	String	否	当缓存类型为Redis时，则不需要设置，保持为空即可。 当缓存引擎为Memcached，且“no_password_access”为“false”时才需要设置，表示通过密码认证访问缓存实例的认证用户名。 由英文字符开头，只能由英文字母、数字、中划线和下划线组成，长度为1~64的字符。 说明 • 当缓存引擎为Memcached时，该参数为可选项。 • 当缓存引擎为Redis时，该参数不需要设置。
vpc_id	String	是	虚拟私有云ID。 获取方法如下： • 方法1：登录虚拟私有云服务的控制台界面，在虚拟私有云的详情页面查找VPC ID。 • 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询VPC列表。
security_group_id	String	是	指定实例所属的安全组。 获取方法如下： • 方法1：登录虚拟私有云服务的控制台界面，在安全组的详情页面查找安全组ID。 • 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询安全组列表。

参数	类型	必选	说明
subnet_id	String	是	子网的网络ID。 获取方法如下： - 方法1：登录虚拟私有云服务的控制台界面，单击VPC下的子网，进入子网详情页面，查找网络ID。 - 方法2：通过虚拟私有云服务的API接口查询，具体操作可参考查询子网列表。
available_zones	Array	是	创建缓存节点到指定且有资源的可用区ID。具体查询方法，请参考 查询可用区信息 ，在查询时，请注意查看该可用区是否有资源。 如果是创建主备、Proxy集群、Cluster集群实例，支持跨可用区部署，可以为备节点指定备可用区。在为节点指定可用区时，用逗号分隔开，具体请查看示例。
product_id	String	是	支持创建的产品的标识。具体查询方法，请参考 查询产品规格列表 。 参数取值格式为：spec_code+ “-” + 计费模式缩写，计费模式缩写取值范围有：h、m、y。其中，h表示计费模式为按需；m表示计费模式为包月；y表示计费模式为包年。 例如：redis.single.au1.large.1-h，表示规格redis.single.au1.large.1的按需计费产品ID。 说明 在查询产品规格中获取product_id时，请务必确保获取的product_id对应的计费模式和您选择的计费模式一致。例如，要订购按需计费、实例类型为Redis3.0的主备实例，可从接口响应消息中，查找charging_type字段为Hourly、spec_code字段为dcs.master_standby的记录对应的product_id。
instance_backup_policy	JSON	否	备份策略，实例类型为主备和集群时支持。具体备份参数，请参考 表8-4 和 表8-5 。
enable_publicip	Boolean	否	Redis缓存实例是否开启公网访问功能。 - true：开启 - false：不开启
publicip_id	String	否	Redis缓存实例绑定的弹性IP地址的id。 如果开启了公网访问功能（即enable_publicip为true），该字段为必选。
enable_ssl	Boolean	否	Redis缓存实例开启公网访问功能时，是否选择支持ssl。 - true：开启 - false：不开启

参数	类型	必选	说明
maintain_begin	String	否	维护时间窗开始时间，格式为HH:mm:ss。 - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为22:00:00、02:00:00、06:00:00、10:00:00、14:00:00和18:00:00。 - 该参数不能单独为空，若该值为空，则结束时间也为空。系统分配一个默认开始时间02:00:00。
maintain_end	String	否	维护时间窗结束时间，格式为HH:mm:ss。 - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 结束时间在开始时间基础上加四个小时，即当开始时间为22:00:00时，结束时间为02:00:00。 - 该参数不能单独为空，若该值为空，则开始时间也为空，系统分配一个默认结束时间06:00:00。
enterprise_project_id	String	否	企业项目ID。
enterprise_project_name	String	否	企业项目名称。
instance_num	Integer	否	表示批量创建缓存实例时，购买的实例个数。仅Redis和Memcached实例支持批量创建。 默认值：1 取值范围：1-100
private_ip	String	否	创建缓存实例手动指定的IP地址,支持Redis和Memcached。
bss_param	JSON	否	表示包周期计费模式的相关参数。如果为空，则默认计费模式为按需计费；如果是包周期方式，请参考 表8-6 设置。
tags	Array of objects	否	实例标签键值。具体请参考 表8-3 。
port	Integer	否	实例自定义端口。只有创建Redis4.0和Redis5.0实例才支持自定义端口，Redis3.0和Memcached实例不支持。 创建Redis4.0和Redis5.0实例，如果没发送该参数或该参数为空，表示实例使用默认端口6379。如果自定义端口，端口范围为1~65535的任意数字。

参数	类型	必选	说明
rename_commands	JSON	否	支持自定义重命名高危命令。只有创建Redis4.0和Redis5.0实例才支持重命名高危命令，Redis3.0和Memcached实例不支持。 创建Redis4.0和Redis5.0实例，如果没发送该参数或该参数为空，表示没有需要重命名的命令。当前支持重命名的高危命令有command、keys、flushdb、flushall和hgetall，其他命令暂不支持重命名。

表 8-3 tags 参数说明

参数	是否必选	参数类型	描述
key	否	String	标签键。 - 不能为空。 - 对于同一资源键值唯一。 - 长度不超过36个字符。
value	否	String	标签值。 - action为create时，必选。action为delete时，非必选。 - 长度不超过43个字符。

表 8-4 备份策略

参数	类型	必选	说明
save_days	Integer	否	当backup_type设置为manual时，该参数为必填。 保留天数，单位：天，取值范围：1-7。
backup_type	String	否	备份类型。 - auto：自动备份 - manual：手动备份 默认值是：manual。
periodical_backup_plan	JSON	是	备份计划，具体请参考 表8-5 。

表 8-5 备份计划

参数	类型	必选	说明
begin_at	String	是	备份执行时间，“00:00-01:00”代表0点开始执行备份。
period_type	String	是	备份周期类型，目前支持“weekly”。
backup_at	Array	是	每周的周几开始备份，取值1-7，1代表周一，7代表周日。
timezone_of_fset	String	否	备份的时区。取值为-1200 ~+1200之间的时区。若为空则默认使用DCS-Server节点的当前时区。

表 8-6 bss_param 参数说明

参数	类型	必选	说明
charging_mode	String	是	功能说明：付费方式（预付费、按需付费；预付费，即包周期付费） 取值范围： • prePaid -预付费，即包年包月； • postPaid -后付费，即按需付费； 默认值是postPaid。 后付费的场景下，bss_param参数的其他字段都会被忽略。
is_auto_pay	String	否	功能说明：下单订购后，是否自动从客户的账户中支付；默认是“不自动支付” 取值范围： • true：是（自动支付，从账户余额自动扣费） • false：否（默认值，只提交订单不支付，需要客户手动去支付） 约束： 自动支付时，只能使用账户的现金支付；如果要使用代金券，请选择不自动支付，然后在用户费用中心，选择代金券支付。 如果没有设置成自动支付，即设置为false时，在创建实例之后，实例状态为“支付中”，用户必须在“费用中心 > 我的订单”，完成订单支付，否则订单一直在支付中，实例没有创建成功。
period_type	String	否	当选择包年包月时，该字段为必选，表示订购资源的周期类型。 取值范围如下： • month：表示包月 • year：表示包年

参数	类型	必选	说明
period_num	Integer	否	功能说明：订购周期数 取值范围：（后续会随运营策略变化） - period_type为month时，为[1,9]， - period_type为year时，为[1,3] 约束：同period_type约束。
is_auto_renew	String	否	当选择包年包月时，该字段为必选，表示是否自动续订资源。 取值范围： - false：不自动续订； - true：自动续订； 默认值为：false 约束： 如果设置为自动续订，到期后，会自动续订一个月（自动续订时间后续可能会变化），详情可联系技术支持咨询。

请求示例

请求URL：

POST https://{dcs_endpoint}/v1.0/{project_id}/instances

- 示例：

创建Redis实例

```
{
  "name": "dcs-APItest",
  "description": "Test",
  "engine": "Redis",
  "engine_version": "3.0",
  "capacity": 2,
  "enterprise_project_id": "0",
  "enterprise_project_name": "default",
  "no_password_access": false,
  "access_user": "",
  "password": "*****",
  "vpc_id": "0402ea19-5457-4032-9d1b-eb48b98f6c66",
  "security_group_id": "4b95a790-5cfc-463d-8de5-42199e55371c",
  "subnet_id": "a3bd29e4-d5bc-414c-a79a-1f35ee4ead88",
  "available_zones": [
    "12c47a78666b4e438cd0c692b9860387"
  ],
  "maintain_begin": "02:00",
  "maintain_end": "06:00",
  "product_id": "dcs.master_standby-h",
  "enable_publicip": false,
  "publicip_id": "",
  "enable_ssl": false,
  "private_ip": "",
  "instance_num": 1
}
```

- 示例

创建按需的Redis主备实例（设置了标签、自定义端口、重命名高危命令、跨可用区部署）

```
{
  "name": "dcs-demo",
  "engine": "Redis",
  "engine_version": "4.0",
  "capacity": 2,
  "no_password_access": true,
  "vpc_id": "c5cd009b-9ba9-41f2-9a26-2b8c3f3ffd39",
  "subnet_id": "7e95c4d6-d7af-424c-a3cf-2be10968fe81",
  "product_id": "redis.ha.au1.large.2-h",
  "available_zones": ["ae04cf9d61544df3806a3feeb401b204",
"d573142f24894ef3bd3664de068b44b0"],
  "security_group_id": "d57fad27-e3e9-43b7-9498-0a698ab63b27",
  "port": "4040",
  "rename_commands":
  {
    "keys": "keys001",
    "flushdb": "flushdb001",
    "flushall": "flushall001"
  },
  "tags": [{
    "key": "dcs001",
    "value": "002"
  },
  {
    "key": "dcs003",
    "value": "004"
  }
]
```

- 示例：

创建包周期的Redis主备实例（设置了备份策略）

```
{
  "name": "dcs-APItest1",
  "description": "",
  "engine": "Redis",
  "engine_version": "3.0.7",
  "capacity": 2,
  "enterprise_project_id": "0",
  "enterprise_project_name": "default",
  "no_password_access": false,
  "access_user": "",
  "password": "*****",
  "vpc_id": "0402ea19-5457-4032-9d1b-eb48b98f6c66",
  "security_group_id": "4b95a790-5cfc-463d-8de5-42199e55371c",
  "subnet_id": "a3bd29e4-d5bc-414c-a79a-1f35ee4ead88",
  "available_zones": [
    "12c47a78666b4e438cd0c692b9860387"
  ],
  "maintain_begin": "02:00",
  "maintain_end": "06:00",
  "product_id": "dcs.master_standby-m",
  "enable_publicip": false,
  "publicip_id": "",
  "enable_ssl": false,
  "instance_num": 1,
  "private_ip": "",
  "instance_backup_policy": {
    "save_days": 1,
    "backup_type": "auto",
    "periodical_backup_plan": {
      "begin_at": "16:00-17:00",
      "period_type": "weekly",
      "backup_at": [
        "1",
        "2",
        "3",

```

```

        "4",
        "5",
        "6",
        "7"
    ],
    "timezone_offset": "+0800"
}
},
"bss_param": {
    "charging_mode": "prePaid",
    "is_auto_pay": "true",
    "period_type": "month",
    "period_num": "1",
    "is_auto_renew": "false"
}
}

```

- 示例：

创建包周期的Redis集群实例（设置了备份策略）

```

{
    "name": "dcs-k3ej",
    "description": "",
    "engine": "Redis",
    "engine_version": "3.0.7",
    "capacity": 64,
    "enterprise_project_id": "0",
    "enterprise_project_name": "default",
    "no_password_access": true,
    "access_user": "",
    "password": "",
    "vpc_id": "0402ea19-5457-4032-9d1b-eb48b98f6c66",
    "security_group_id": "4b95a790-5cfc-463d-8de5-42199e55371c",
    "subnet_id": "a3bd29e4-d5bc-414c-a79a-1f35ee4ead88",
    "available_zones": [
        "12c47a78666b4e438cd0c692b9860387"
    ],
    "maintain_begin": "02:00",
    "maintain_end": "06:00",
    "product_id": "dcs.cluster-m",
    "enable_publicip": false,
    "publicip_id": "",
    "enable_ssl": false,
    "instance_num": 1,
    "private_ip": "",
    "instance_backup_policy": {
        "save_days": 1,
        "backup_type": "auto",
        "periodical_backup_plan": {
            "begin_at": "16:00-17:00",
            "period_type": "weekly",
            "backup_at": [
                "1",
                "2",
                "3",
                "4",
                "5",
                "6",
                "7"
            ]
        },
        "timezone_offset": "+0800"
    }
},
"bss_param": {
    "charging_mode": "prePaid",
    "is_auto_pay": "true",
    "period_type": "month",
    "period_num": "1",
    "is_auto_renew": "false"
}
}

```

```
    }
  }
}

• 示例：
  创建Memcached主备实例（ 设置了备份策略）

{
  "name": "dcs-a11b",
  "description": "Create an instance",
  "engine": "Memcached",
  "engine_version": "",
  "capacity": 2,
  "password": "XXXXXX",
  "vpc_id": "27d99e17-42f2-4751-818f-5c8c6c03ff16",
  "security_group_id": "1477393a-29c9-4de5-843f-18ef51257c8e",
  "subnet_id": "ec2f34b9-20eb-4872-85bd-bea9fc943129",
  "available_zones": [
    "1d7b939b382c4c3bb3481a8ca10da769"
  ],
  "product_id": "XXXXXX",
  "instance_backup_policy": {
    "save_days": 1,
    "backup_type": "auto",
    "periodical_backup_plan": {
      "begin_at": "00:00-01:00",
      "period_type": "weekly",
      "backup_at": [
        1,
        2,
        3,
        4,
        5,
        6,
        7
      ]
    }
  },
  "maintain_begin": "22:00:00",
  "maintain_end": "02:00:00",
  "no_password_access": "true",
  "access_user": "",
  "timezone_offset": ""
}
```

响应消息

响应参数

参数说明见[表8-7](#)。

表 8-7 参数说明

参数	类型	说明
instance_id	String	缓存实例ID。 如果批量创建实例，则会返回多个实例ID，以逗号分隔。 说明 该参数用于兼容老版本，参数列出的实例列表与 instances 相同。
instances	JSON	缓存实例列表。具体参数，请参考 表8-8 。

表 8-8 instances 参数说明

参数	类型	说明
instance_id	String	缓存实例ID。
instance_name	String	缓存实例名称。

响应示例

如果是按需实例，响应消息如下：

```
{
  "instances": [
    {
      "instance_id": "3c49fd6b-fc7c-419e-9644-b6cce008653f",
      "instance_name": "dcs-test005"
    }
  ],
  "instance_id": "3c49fd6b-fc7c-419e-9644-b6cce008653f"
}
```

如果是包周期实例，响应消息如下：

```
{
  "instance_id": "77284d69-cd51-4bf1-aa30-2e60f055800c",
  "order_id": "CS2005091653L2DAM",
  "instances": [
    {
      "instance_id": "77284d69-cd51-4bf1-aa30-2e60f055800c",
      "instance_name": "dcs-APITest1"
    }
  ]
}
```

状态码

操作成功的状态码如[表8-9](#)所示，其他响应见[表7-1](#)。

表 8-9 状态码

状态码	描述
200	创建缓存实例成功。

8.1.2 删除实例

功能介绍

删除指定的缓存实例，释放该实例的所有资源。

 说明

如果是删除按需资源，请按照本章节执行；如果是删除包周期资源，即退订，请参考[退订包周期资源](#)。

URI

DELETE /v1.0/{project_id}/instances/{instance_id}

参数说明见[表8-10](#)。

表 8-10 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

无。

请求示例

请求URL:

DELETE https://{*dc*s_endpoint}/v1.0/{project_id}/instances/{instance_id}

响应消息

响应参数

无。

响应示例

无。

状态码

操作成功的状态码如[表8-11](#)所示，其他响应见[表7-1](#)。

表 8-11 状态码

状态码	描述
204	删除缓存实例成功。

8.1.3 批量删除实例

功能介绍

批量删除多个缓存实例。

URI

DELETE /v1.0/{project_id}/instances?allFailure={allFailure}

参数说明见[表8-12](#)。

表 8-12 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
allFailure	String	否	是否批量删除创建失败的缓存实例。 取值如下： • true，表示删除租户所有创建失败的缓存实例，此时请求参数instances可为空； • false或者其他值，表示删除instances参数数组中指定的缓存实例。

请求消息

请求参数

参数说明见[表8-13](#)。

表 8-13 参数说明

参数	类型	必选	说明
instances	Array	否	实例的ID列表。 仅当URI中参数allFailure值为“false”或者其他值时，才需要配置该参数。 一次最多可删除50个实例。

请求URL:

DELETE https://{dcs_endpoint}/v1.0/{project_id}/instances?allFailure={allFailure}

当allFailure为false时，请求示例如下：

```
{
  "instances": [
    "54602a9d-5e22-4239-9123-77e350df4a34",
    "7166cdea-dbad-4d79-9610-7163e6f8b640"
  ]
}
```

响应消息

响应参数

当URI中参数allFailure值为true时，响应返回为空。当URI中参数allFailure值为false时，参数说明见[表8-14](#)。

表 8-14 参数说明

参数	类型	说明
results	Array	删除实例的结果，具体请参考 表8-15 。

表 8-15 results 参数说明

参数	类型	说明
instance	String	缓存实例ID。
result	String	操作结果，取值有success或failed。

响应示例

```
{
  "results": [
    {
      "instance": "54602a9d-5e22-4239-9123-77e350df4a34",
      "result": "success"
    },
    {
      "instance": "7166cdea-dbad-4d79-9610-7163e6f8b640",
      "result": "success"
    }
  ]
}
```

状态码

操作成功的状态码如[表8-16](#)所示，其他响应见[表7-1](#)。

表 8-16 状态码

状态码	描述
200	批量删除缓存实例成功。
204	清理创建失败的缓存实例成功。

8.1.4 查询指定实例

功能介绍

通过实例ID查询实例的详细信息。

URI

GET /v1.0/{project_id}/instances/{instance_id}

参数说明见[表8-17](#)。

表 8-17 参数说明

参数	类型	必选	说明
project_id	String	是	获取方式，参考 获取项目ID 。
instance_id	String	是	待查询的实例ID。

请求消息

请求参数

无。

请求示例

请求URL：

GET https://{*dc*s_endpoint}/v1.0/{project_id}/instances/{instance_id}

响应消息

响应参数

参数说明见[表8-18](#)。

表 8-18 参数说明

参数	类型	说明
name	String	实例名称。
engine	String	缓存实例的引擎类型。
capacity	Integer	缓存实例的容量（ G Byte ）。
ip	String	连接缓存实例的IP地址。如果是集群实例，返回多个IP地址，使用逗号分隔。如：192.168.0.1，192.168.0.2。
domainName	String	实例的域名。
port	Integer	缓存的端口。
status	String	缓存实例的状态。详细状态说明见 缓存实例状态说明 。
libos	Boolean	- true：该实例是libos实例。 - false：该实例不是libos实例。

参数	类型	说明
description	String	实例描述。
max_memory	Integer	总内存，单位：MB。
used_memory	Integer	已使用的内存，单位：MB。
instance_id	String	实例ID。
resource_spec_code	String	资源规格标识。对应控制台创建缓存实例页面的实例规格名称。
engine_version	String	缓存版本。
internal_version	String	DCS内部版本号。
charging_mode	Integer	付费模式，0表示按需计费，1表示包年/包月计费。
capacity_minor	String	单机小规格的缓存容量。
vpc_id	String	VPC ID。
vpc_name	String	VPC的名称。
created_at	String	完成创建时间。格式为： 2017-03-31T12:24:46.297Z
error_code	String	实例创建失败或状态异常时的错误码，错误码说明见 表8-19 。
user_id	String	用户id。
user_name	String	用户名。
maintain_begin	String	维护时间窗开始时间，格式为HH:mm:ss。
maintain_end	String	维护时间窗结束时间，格式为HH:mm:ss。
no_password_access	String	是否允许免密码访问缓存实例。 • true：该实例无需密码即可访问。 • false：该实例必须通过密码认证才能访问。
access_user	String	通过密码认证访问缓存实例的认证用户名。
enable_publicip	Boolean	Redis缓存实例是否开启公网访问功能。 • true：开启 • false：不开启
publicip_id	String	Redis缓存实例绑定的弹性IP地址的id。 如果未开启公网访问功能，该字段值为null。
publicip_address	String	Redis缓存实例绑定的弹性IP地址。 如果未开启公网访问功能，该字段值为null。

参数	类型	说明
enable_ssl	Boolean	Redis缓存实例开启公网访问功能时，是否选择支持ssl。 - true：开启 - false：不开启
service_upgrade	Boolean	实例是否存在升级任务。 - true：存在 - false：不存在
service_task_id	String	升级任务的ID。 - 当service_upgrade为true时，该参数为升级任务的ID。 - 当service_upgrade为false时，该参数为空。
enterprise_project_id	String	企业项目ID。
available_zones	Array	实例节点所在的可用区。返回“可用区ID”。
subnet_id	String	子网ID。
security_group_id	String	安全组ID。
backend_addrs	String	集群实例的后端服务地址。
product_id	String	产品标识。
security_group_name	String	租户安全组名称。
subnet_name	String	子网名称。
subnet_cidr	String	子网网段。
order_id	String	订单ID。
instance_backup_policy	JSON	备份策略，实例类型为主备和集群时支持。详情参考 表8-4 和 表8-5 。
tags	Array of objects	实例标签键值。

表 8-19 错误码说明

错误码	说明
dcs.00.0007	系统异常。
dcs.01.0001	内部服务错误。

错误码	说明
dcs.01.0002	内部服务错误。
dcs.01.0003	内部服务错误。
dcs.02.0001	VPC创建失败。
dcs.02.0002	VPC创建失败。
dcs.02.0003	安全组创建失败。
dcs.02.0004	子网创建失败。
dcs.02.0005	子网状态异常。
dcs.03.0001	创建ECS失败。
dcs.03.0002	创建ECS失败。
dcs.03.0003	创建ECS失败。
dcs.03.0004	创建ECS失败。
dcs.03.0005	ECS绑定网卡失败。
dcs.03.0007	ECS启动失败。
dcs.03.0008	ECS启动失败。
dcs.03.0009	ECS停止失败。
dcs.03.0017	实例部分节点故障。
dcs.04.0002	实例部署异常。
dcs.04.0003	无法连接实例。
dcs.04.0004	存在两个主节点，主备节点间网络连接可能有问题。

响应示例

```
{
  "name": "dcs-125d",
  "engine": "Redis",
  "capacity": 64,
  "ip": "192.168.1.48",
  "domainName": "XXXXXX",
  "port": 6379,
  "status": "RUNNING",
  "libos": false,
  "description": null,
  "task": null,
  "max_memory": 65536,
  "used_memory": 33,
  "instance_id": "979e9a6b-e2f5-4b5d-8e98-fea495ea27a1",
  "resource_spec_code": "dcs.cluster",
  "engine_version": "3.0",
  "internal_version": null,
  "charging_mode": 0,
  "capacity_minor": null,
  "vpc_id": "d1e8008d-8182-4c86-a142-1ae9349bdc64",
```

```
"vpc_name": "vpc-807d",
"task_status": null,
"created_at": "2018-12-11T07:24:18.149Z",
"error_code": null,
"user_id": "9bfac09d9bcd4eb7bb1465a7bcff815b",
"user_name": "r00209145",
"maintain_begin": "02:00:00",
"maintain_end": "06:00:00",
"no_password_access": "true",
"access_user": null,
"enable_publicip": false,
"publicip_id": null,
"publicip_address": null,
"enable_ssl": false,
"service_upgrade": false,
"service_task_id": null,
"is_free": null,
"enterprise_project_id": "0",
"available_zones": ["f84448fd537f46078dd8bd776747f573"],
"subnet_id": "9b32218f-6389-485a-912e-076fe9df6fa7",
"security_group_id": "ecefda9-c8be-42aa-9d5d-9ab46a221240",
"backend_addrs": ["192.168.1.136:6379",
"192.168.1.224:6379"],
"product_id": "00301-17063-0--0",
"security_group_name": "sg--test-hanwang",
"subnet_name": "subnet-7e46",
"order_id": null,
"subnet_cidr": "192.168.1.0/24",
"instance_backup_policy": null,
"tags": [
  {
    "key": "dcs001",
    "value": "001"
  },
  {
    "key": "dcs002",
    "value": "002"
  }
],
"enterprise_project_name": null
}
```

状态码

操作成功的状态码如[表8-20](#)所示，其他响应见[表7-1](#)。

表 8-20 状态码

状态码	描述
200	查询指定实例成功。

8.1.5 查询所有实例列表

功能介绍

查询租户的缓存实例列表，支持按照条件查询。

URI

GET /v1.0/{project_id}/instances?
start={start}&limit={limit}&name={name}&status={status}&id={id}&include_failure={include_failure}&isExactMatchName={isExactMatchName}&ip={ip}&tags={key}={value}

参数说明见[表8-21](#)。

表 8-21 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
start	Integer	否	数据的起始序号，默认起始序号为1。
limit	Integer	否	每页显示条数，最小值为1，最大值为1000，若不设置该参数，则为最大显示条数1000。
name	String	否	实例名称。
id	String	否	实例ID。
status	String	否	实例状态。详细状态说明见 缓存实例状态说明 。
ip	String	否	连接缓存实例的IP地址。
include_failure	String	否	是否返回创建失败的实例数。 当参数值为“true”时，返回创建失败的实例数。参数值为“false”或者其他值，不返回创建失败的实例数。
isExactMatchName	String	否	是否按照实例名称进行精确匹配查询。 默认为“false”，表示模糊匹配实例名称查询。若参数值为“true”表示按照实例名称进行精确匹配查询。
tags	Array of objects	否	根据实例标签键值对进行查询。{key}表示标签键，{value}表示标签值。 如果同时使用多个标签键值对进行查询，中间使用逗号分隔开，表示查询同时包含指定标签键值对的实例。

示例

GET https://{[dcs_endpoint](#)}/v1.0/bd6b78e2ff9e4e47bc260803ddcc7a21/instances?
start=1&limit=10&name=&status=&id=&include_failure=true&isExactMatchName=false

或：

GET https://{[dcs_endpoint](#)}/v1.0/bd6b78e2ff9e4e47bc260803ddcc7a21/instances?
tags=dc001=001,dc002=002

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-22](#)。

表 8-22 参数说明

参数	类型	说明
instances	Array	实例的详情数组。
instance_num	Integer	实例个数。

表 8-23 instance 参数说明

参数	类型	说明
name	String	实例名称。
engine	String	缓存引擎。
capacity	Integer	缓存容量（ G Byte ）。
ip	String	连接缓存实例的IP地址。如果是集群实例，返回多个IP地址，使用逗号分隔。如：192.168.0.1，192.168.0.2。
domainName	String	实例的域名。
port	Integer	缓存的端口。
status	String	缓存实例的状态。详细状态说明见 缓存实例状态说明 。
max_memory	Integer	总内存，单位：MB。
used_memory	Integer	已使用的内存，单位：MB。
instance_id	String	实例ID。
resource_spec_code	String	资源规格标识。对应控制台创建缓存实例页面的实例规格名称。
engine_version	String	缓存版本。
internal_version	String	DCS内部版本号。

参数	类型	说明
charging_mode	Integer	付费模式，0表示按需计费，1表示包年/包月计费。
capacity_minor	String	小规格缓存容量（ G Byte ）。
vpc_id	String	VPC ID。
vpc_name	String	VPC的名称。
created_at	String	创建时间。格式为： 2017-03-31T12:24:46.297Z
error_code	String	实例创建失败或状态异常时的错误码，错误码的说明见 表8-19 。
user_id	String	用户id。
user_name	String	用户名。
maintain_begin	String	维护时间窗开始时间，格式为HH:mm:ss。
maintain_end	String	维护时间窗结束时间，格式为HH:mm:ss。
no_password_access	String	是否允许免密码访问缓存实例。 • true：该实例无需密码即可访问。 • false：该实例必须通过密码认证才能访问。
access_user	String	通过密码认证访问缓存实例的认证用户名。
enable_publicip	Boolean	Redis缓存实例是否开启公网访问功能。 • true：开启 • false：不开启
publicip_address	String	Redis缓存实例绑定的弹性IP地址。 如果未开启公网访问功能，该字段值为null。
publicip_id	String	Redis缓存实例绑定的弹性IP地址的id。 如果未开启公网访问功能，该字段值为null。
enable_ssl	Boolean	Redis缓存实例开启公网访问功能时，是否选择支持ssl。 • true：开启 • false：不开启
service_upgrade	Boolean	实例是否存在升级任务。 • true：存在 • false：不存在

参数	类型	说明
service_task_id	String	升级任务的ID。 - 当service_upgrade为true时，该参数为升级任务的ID。 - 当service_upgrade为false时，该参数为空。
enterprise_project_id	String	企业项目ID。
security_group_id	String	租户安全组名称。
tags	Array of objects	实例标签键值。

响应示例

```
{
  "instances": [
    {
      "name": "dcs-lxy",
      "engine": "Redis",
      "capacity": 4,
      "ip": "192.168.7.146",
      "domainName": null,
      "port": 6379,
      "status": "CREATING",
      "max_memory": 3277,
      "used_memory": 0,
      "instance_id": "a4d31cb6-3d72-4fdc-8ec9-6e3a41e47f71",
      "resource_spec_code": "dcs.master_standby",
      "engine_version": "3.0",
      "internal_version": null,
      "charging_mode": 0,
      "capacity_minor": null,
      "vpc_id": "c71d9731-9b0c-43e9-ab2a-716af9d9fd55",
      "vpc_name": "CCE-AutoCreate-VPC-7qvs1",
      "created_at": "2019-09-23T02:40:06.123Z",
      "error_code": null,
      "user_id": "50a4156d334a4a82b8745dc730dc1e00",
      "user_name": "hwstaff_f00443635",
      "maintain_begin": "02:00:00",
      "maintain_end": "06:00:00",
      "no_password_access": "true",
      "access_user": null,
      "enable_publicip": false,
      "publicip_id": null,
      "publicip_address": null,
      "enable_ssl": false,
      "service_upgrade": false,
      "service_task_id": "",
      "enterprise_project_id": "0",
      "tags": [
        {
          "key": "dcs001",
          "value": "001"
        },
        {
          "key": "dcs002",
          "value": "002"
        }
      ],
      "security_group_id": "0cc8fdb7-872a-49da-a062-88ccc39463b5"
    }
  ]
}
```

```
    },  
    "instance_num": 1  
}
```

状态码

操作成功的状态码如表8-24所示，其他响应见表7-1。

表 8-24 状态码

状态码	描述
200	查询所有实例列表成功。

8.1.6 修改实例信息

功能介绍

修改缓存实例的信息，可修改信息包括实例名称、描述、备份策略、维护时间窗开始和结束时间以及安全组。

URI

PUT /v1.0/{project_id}/instances/{instance_id}

参数说明见表8-25。

表 8-25 参数说明

参数	类型	必选	备注
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

参数说明见表8-26。

表 8-26 参数说明

参数	类型	必选	说明
name	String	否	实例名称。 由英文字符开头，只能由英文字母、数字、中划线和下划线组成。 创建单个实例时，名称长度为4到64位的字符串。批量创建实例时，名称长度为4到56位的字符串，且实例名称格式为“自定义名称-n”，其中n从000开始，依次递增。例如，批量创建两个实例，自定义名称为dcs_demo，则两个实例的名称为dcs_demo-000和dcs_demo-001。
description	String	否	实例的描述信息。 长度不超过1024的字符串。 说明 \与"在json报文中属于特殊字符，如果参数值中需要显示\或者"字符，请在字符前增加转义字符\\，比如\\或者\"。
instance_backup_policy	JSON	否	备份策略，实例类型为主备和集群时支持。详情参考 表8-4 和 表8-5 。
maintain_begin	String	否	维护时间窗开始时间，格式为HH:mm:ss。 - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 开始时间必须为22:00:00、02:00:00、06:00:00、10:00:00、14:00:00和18:00:00。 - 该参数不能单独为空，若该值为空，则结束时间也为空。
maintain_end	String	否	维护时间窗结束时间，格式为HH:mm:ss。 - 维护时间窗开始和结束时间必须为指定的时间段，可参考查询维护时间窗时间段获取。 - 结束时间在开始时间基础上加四个小时，即当开始时间为22:00:00时，结束时间为02:00:00。 - 该参数不能单独为空，若该值为空，则开始时间也为空。
security_group_id	String	否	安全组ID。 可从虚拟私有云服务的控制台界面或者API接口查询得到。

请求示例

请求URL：

PUT https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}

- 示例1：

```
{
  "description": "instance description"
}
```
- 示例2：

```
{
  "name": "dcs002",
  "description": "instance description",
  "instance_backup_policy": {
    "backup_type": "auto",
    "save_days": 1,
    "periodical_backup_plan": {
      "begin_at": "00:00-01:00",
      "period_type": "weekly",
      "backup_at": [
        "1",
        "2",
        "3",
        "4",
        "6",
        "7"
      ]
    }
  },
  "security_group_id": "18e9309f-f81a-4749-bb21-f74576292162",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00"
}
```

响应消息

响应参数

无。

响应样例

无。

状态码

操作成功的状态码如[表8-27](#)所示，其他响应见[表7-1](#)。

表 8-27 状态码

状态码	描述
204	修改缓存实例成功。

8.1.7 扩容缓存实例

功能介绍

用户可以为状态为“运行中”的Redis和Memcached缓存实例进行扩容和实例类型从单机升级为主备操作。

URI

POST /v1.0/{project_id}/instances/{instance_id}/extend

参数说明见[表8-28](#)。

表 8-28 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

参数说明见[表8-29](#)。

表 8-29 参数说明

参数	类型	必选	说明
new_capacity	Integer	是	新的缓存实例规格，新的规格必须大于扩容前的规格，单位：GB。 取值必须是当前产品支持的实例规格，具体取值，请查看《产品介绍》手册的 实例规格 章节，查找到对应的“内存规格（GB）”列。
spec_code	String	是	扩容后的缓存实例产品规格编码。 如果是Redis3.0实例，该字段为可选。 如果是Redis4.0和Redis5.0实例，该字段为必选。 具体取值，请查看《产品介绍》手册的 实例规格 章节，查找到对应的“产品规格名称”列。

请求示例

POST https://{*dcs_endpoint*}/v1.0/{project_id}/instances/{instance_id}/extend

```
{
  "new_capacity": 4,
  "spec_code": "redis.single.au1.large.4"
}
```

响应消息

响应参数

无

响应示例

无

状态码

操作成功的状态码如[表8-30](#)所示，其他响应见[表7-1](#)。

表 8-30 状态码

状态码	描述
204	扩容缓存实例成功。

8.2 实例管理

8.2.1 重启实例或清空数据

功能介绍

重启运行中的DCS缓存实例。

清空Redis4.0/Redis5.0的实例数据，数据清空后，无法撤销，且无法恢复，请谨慎操作。

URI

PUT /v1.0/{project_id}/instances/status

参数说明见[表8-31](#)。

表 8-31 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。

请求消息

请求参数

参数说明见[表8-32](#)。

表 8-32 参数说明

参数	类型	必选	说明
action	String	是	对实例的操作：restart、flush。 说明 当前版本，只有DCS2.0实例支持清空数据功能，即flush操作。
instances	Array	是	实例的ID列表。

请求示例

```
PUT https://{dcs_endpoint}/v1.0/{project_id}/instances/status

{
  "action": "restart",
  "instances": [
    "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef"
  ]
}
```

响应消息

响应参数

参数说明见[表8-33](#)。

表 8-33 参数说明

参数	类型	说明
results	Array	修改实例的结果。

表 8-34 results 参数说明

参数	类型	说明
instance	String	缓存实例ID。
result	String	操作结果：success、failed。

响应示例

```
{
  "results": [
    {
      "result": "success",
      "instance": "2e803f66-fbb0-47ad-b6cb-fb87f5bed4ef"
    }
  ]
}
```

状态码

操作成功的状态码如表8-35所示，其他响应见表7-1。

表 8-35 状态码

状态码	描述
200	重启实例或清除实例数据成功。

8.2.2 查询分片信息

功能介绍

查询读写分离实例和集群实例的分片和副本信息，其中，读写分离实例仅Redis4.0和Redis5.0的主备实例支持。

 说明

该接口当前仅在华北-北京四区域开放。

URI

GET /v1.0/{project_id}/instance/{instance_id}/groups

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
instance_id	是	String	实例ID。

请求消息

请求参数

无

请求示例

请求URL：

```
GET https://{dcs_endpoint}/v1.0/{project_id}/instance/{instance_id}/groups
```

响应消息

响应参数

表 8-36 响应参数

参数	参数类型	描述
group_count	String	实例分片总数。
group_list	Array of objects	分片列表。具体参数请参见表8-37。

表 8-37 group_list 参数说明

参数	参数类型	描述
group_id	String	分片id
group_name	String	分片名称
replication_list	Array of objects	每个分片包含的副本列表。具体参数请参见表8-38。

表 8-38 replication_list 参数说明

参数	参数类型	描述
replication_ip	String	副本IP。
replication_role	String	副本角色，取值有： ● master：表示主节点。 ● slave：表示从节点。
node_id	String	节点ID。
replication_id	String	副本ID。
is_replication	Boolean	是否是新加副本。
status	String	副本状态。

响应示例

状态码为 200 时:

```
{
  "group_list": [
    {
      "group_id": "a11d3ba3-7898-4371-bfbe-783ab9280cc7",
      "group_name": "group-2",
      "replication_list": [
        {
          "replication_ip": "192.168.0.156",
          "replication_role": "master",
          "node_id": "8aa6999f6de47c69016df0e7de1c23de",
          "replication_id": "1a5f2b63-eedc-4bb1-a25f-262b5d604a1f",
          "is_replication": false,

```

```
        "status": "Active"
      },
      {
        "replication_ip": "192.168.0.45",
        "replication_role": "slave",
        "node_id": "8aa6999f6de47c69016df0e7de2e23df",
        "replication_id": "2b892b07-dfe8-4b86-98cc-736b4e8e420f",
        "is_replication": false,
        "status": "Active"
      }
    ]
  },
  {
    "group_id": "2ba84ddc-558c-4465-bfe9-dbfaac024565",
    "group_name": "group-1",
    "replication_list": [
      {
        "replication_ip": "192.168.0.95",
        "replication_role": "master",
        "node_id": "8aa6999f6de47c69016df0e7ddf623dc",
        "replication_id": "b46cbcca-05a7-47b0-bd01-df768a1faf73",
        "is_replication": false,
        "status": "Active"
      },
      {
        "replication_ip": "192.168.0.204",
        "replication_role": "slave",
        "node_id": "8aa6999f6de47c69016df0e7de0923dd",
        "replication_id": "56ddd52e-0c95-40f2-a596-b578456c1431",
        "is_replication": false,
        "status": "Active"
      }
    ]
  },
  {
    "group_id": "995e1f3b-f9c0-4acf-ad9d-89123d1fd12a",
    "group_name": "group-0",
    "replication_list": [
      {
        "replication_ip": "192.168.0.233",
        "replication_role": "master",
        "node_id": "8aa6999f6de47c69016df0e7ddd223da",
        "replication_id": "71749f79-e9ea-4173-8bfa-b97357153af8",
        "is_replication": false,
        "status": "Active"
      },
      {
        "replication_ip": "192.168.0.27",
        "replication_role": "slave",
        "node_id": "8aa6999f6de47c69016df0e7dde423db",
        "replication_id": "707b7318-9766-47a8-91ad-7d0feab64229",
        "is_replication": false,
        "status": "Active"
      }
    ]
  }
],
"group_count": 3
}
```

返回值

返回值	描述
200	查询分片信息成功

错误码

无

8.2.3 添加副本

功能介绍

为Cluster集群实例的分片添加副本。

说明

该接口当前仅在华北-北京四区域开放。

URI

POST /v1.0/{project_id}/instance/{instance_id}/groups/{group_id}/replications

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
instance_id	是	String	实例ID。
group_id	是	String	分片ID。即 查询分片信息 响应消息中的group_id参数的值。

URI示例:

POST https://{dcs_endpoint}/v1.0/{project_id}/instance/{instance_id}/groups/{group_id}/replications

请求消息

请求参数

参数	是否必选	参数类型	描述
repl_az_id	否	String	表示指定副本所在的可用区ID。 可用区ID可通过 查询可用区信息 接口查询，可用区必须是有资源的，否则添加失败。

请求示例

```
{
  "repl_az_id": "12c47a78666b4e438cd0c692b9860387"
}
```

响应消息

无

返回值

返回值	描述
204	添加副本成功。

8.2.4 查询运行中实例的统计信息

功能介绍

查询当前租户下处于“运行中”状态的缓存实例的统计信息。

URI

GET /v1.0/{project_id}/instances/statistic

参数说明见[表8-39](#)。

表 8-39 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。

请求消息

请求参数

无。

请求示例

GET https://{dcs_endpoint}/v1.0/{project_id}/instances/statistic

响应消息

响应参数

参数说明见[表8-40](#)。

表 8-40 参数说明

参数	类型	说明
statistics	Array	该租户下处于“运行中”状态的实例的统计信息，具体请参考 表8-41 。

表 8-41 statistics 参数说明

参数	类型	说明
keys	Integer	缓存存储的数据条数。
instance_id	String	实例ID。
used_memory	Integer	缓存已经使用内存，单位：MB。
max_memory	Integer	缓存的总内存，单位：MB。
cmd_get_count	Integer	缓存get命令被调用次数。
cmd_set_count	Integer	缓存set命令被调用次数。
used_cpu	String	CPU使用率，单位：百分比。
input_kbps	String	缓存实例网络入流量，单位：KB/s。
output_kbps	String	缓存实例网络出流量，单位：KB/s。

响应示例

```
{
  "statistics": [{
    "keys": 0,
    "instance_id": "e008652d-18e0-43ff-924e-072261e0372a",
    "used_memory": 0,
    "max_memory": 460,
    "cmd_get_count": 0,
    "cmd_set_count": 0,
    "used_cpu": "0.0",
    "input_kbps": "0.0",
    "output_kbps": "0.0"
  }, {
    "keys": 0,
    "instance_id": "c577a1eb-33b7-42c7-8231-ad32358599ac",
    "used_memory": 0,
    "max_memory": 460,
    "cmd_get_count": 0,
    "cmd_set_count": 0,
    "used_cpu": "0.0",
    "input_kbps": "0.0",
    "output_kbps": "0.0"
  }, {
    "keys": 0,
    "instance_id": "e8b98471-55d5-4695-b0bb-8f336a98e207",
    "used_memory": 0,
    "max_memory": 460,
    "cmd_get_count": 0,
    "cmd_set_count": 0,
    "used_cpu": "0.0",
    "input_kbps": "0.03",
    "output_kbps": "1.19"
  }, {
    "keys": 0,
    "instance_id": "bc61c690-4b34-4cbe-9ce3-11246aea7aba",
    "used_memory": 0,
    "max_memory": 6963,
    "cmd_get_count": 0,
    "cmd_set_count": 0,
  }
}
```

```
      "used_cpu": "0.0",
      "input_kbps": "0.0",
      "output_kbps": "0.0"
    }
  ]
}
```

状态码

操作成功的状态码如表8-42所示，其他响应见表7-1。

表 8-42 状态码

状态码	描述
200	查询实例统计信息成功。

8.2.5 查询实例状态

功能介绍

查询该租户在当前区域下不同状态的实例数。

URI

GET /v1.0/{project_id}/instances/status?includeFailure={includeFailure}

参数说明见表8-43。

表 8-43 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
includeFailure	String	否	是否返回创建失败的实例数。 - 当参数值为“true”时，返回的统计包括创建失败的实例数。 - 参数值为“false”或者其他值，返回的统计不包括创建失败的实例数。

请求消息

请求参数

无。

请求示例

```
GET https://{dcs_endpoint}/v1.0/{project_id}/instances/status?includeFailure=true
```


响应消息

响应参数

参数说明见[表8-44](#)。

表 8-44 参数说明

参数	类型	说明
creating_count	Integer	正在创建的实例数。
frozen_count	Integer	已冻结的实例数。
deleting_count	Integer	正在删除的实例数。
running_count	Integer	正在运行的实例数。
error_count	Integer	异常的实例数。
restarting_count	Integer	正在重启的实例数。
createfailed_count	Integer	创建失败的实例数。
extending_count	Integer	扩容中的实例数。

响应示例

```
{
  "extending_count": 0,
  "creating_count": 0,
  "deleting_count": 0,
  "running_count": 16,
  "error_count": 0,
  "frozen_count": 0,
  "restarting_count": 0,
  "createfailed_count": 44
}
```

状态码

操作成功的状态码如[表8-45](#)所示，其他响应见[表7-1](#)。

表 8-45 状态码

状态码	描述
200	查询实例数量统计信息成功。

8.2.6 修改密码

功能介绍

修改缓存实例的密码。

URI

PUT /v1.0/{project_id}/instances/{instance_id}/password

参数说明见[表8-46](#)。

表 8-46 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

参数说明见[表8-47](#)。

表 8-47 参数说明

参数	类型	必选	说明
old_password	String	是	旧密码。
new_password	String	是	新密码。 实例密码复杂度要求： • 密码不能为空。 • 密码不能和账号及账号的倒序相同。 • 密码长度在8到32位之间。 • 至少必须包含如下四种字符中的三种： - 小写字母 - 大写字母 - 数字 - 特殊字符包括 (`~!@#\$%^&*()-_+=+\\{};,<.>/?)

请求示例

PUT https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/password

```
{
  "old_password": "XXXXXX",
  "new_password": "XXXXXX"
}
```

响应消息

响应参数

参数说明见[表8-48](#)。

表 8-48 参数说明

参数	类型	说明
result	String	密码修改结果： - 成功：success； - 密码验证失败：passwordFailed； - 已锁定：locked； - 失败：failed。
message	String	修改结果。
retry_times_left	String	密码验证剩余次数。验证失败时该参数返回不为null。
lock_time	String	锁定时间。验证失败时和锁定时该参数返回不为null。
lock_time_left	String	锁定剩余时间。锁定时该参数返回不为null。

响应示例

```
//Change password successful.
{
  "result" : "success",
  "message" : "Modify DCSInstance password success.",
  "retry_times_left" : "5",
  "lock_time" : "0",
  "lock_time_left" : "0"
}
//Change password failed.
{
  "result" : "passwordFailed",
  "message" : "verify password failed.",
  "retry_times_left" : "4",
  "lock_time" : "5",
  "lock_time_left" : "5"
}
```

状态码

操作成功的状态码如[表8-49](#)所示，其他响应见[表7-1](#)。

表 8-49 状态码

状态码	描述
200	修改密码成功。

8.3 参数管理

8.3.1 修改实例配置参数

功能介绍

为了确保分布式缓存服务发挥出最优性能，您可以根据自己的业务情况对DCS缓存实例的运行参数进行调整。

URI

PUT /v1.0/{project_id}/instances/{instance_id}/configs

参数说明见[表8-50](#)。

表 8-50 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID
instance_id	String	是	待修改的实例ID

请求消息

请求参数

参数说明见[表8-51](#)。

表 8-51 参数说明

参数	类型	必选	说明
redis_config	Array	是	实例配置项数组。

表 8-52 redis_config 参数说明

字段	类型	必选	说明
param_id	String	是	实例配置项ID。
param_name	String	是	实例配置项名。
param_value	String	是	实例配置项的值。

对于[表8-52](#)的参数值要求可参考[表8-57](#)。

请求示例

```
PUT https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/configs
{
  "redis_config": [
```

```
{
  {
    "param_id": "1",
    "param_name": "timeout",
    "param_value": "100"
  }
}
```

响应消息

响应参数

无。

响应样例

无。

状态码

操作成功的状态码如[表8-53](#)所示，其他响应见[表7-1](#)。

表 8-53 状态码

状态码	描述
204	修改实例参数成功。

8.3.2 查询实例配置参数

功能介绍

查询指定实例的配置参数信息。

URI

GET /v1.0/{project_id}/instances/{instance_id}/configs

参数说明见[表8-54](#)。

表 8-54 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	待查询的实例ID。

请求消息

请求参数

无。

请求示例

```
GET https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/configs
```

响应消息

响应参数

参数说明见[表8-55](#)。

表 8-55 参数说明

参数	类型	说明
status	String	实例运行状态。
instance_id	String	实例ID。
redis_config	Array	实例配置项数组。具体请参考 表8-56 。
config_status	String	实例修改状态。 • UPDATING • FAILURE • SUCCESS
config_time	String	实例操作时间。格式为： 2017-03-31T12:24:46.297Z

表 8-56 redis_config 参数说明

参数	类型	说明
description	String	配置项的描述。
param_id	Integer	配置参数ID。取值见 表8-57 “参数ID”列。
param_name	String	配置参数名称。取值见 表8-57 “参数名称”列。
param_value	String	配置参数值。
default_value	String	配置参数的默认值。取值见 表8-57 “默认值”列。
value_type	String	配置参数的值类型。取值见 表8-57 “参数类型”列。
value_range	String	配置参数的取值范围。取值见 表8-57 “取值范围”列。

缓存实例配置参数的说明见[表8-57](#)。

表 8-57 缓存实例配置参数说明

参数 ID	参数名称	适用引擎	参数类型	参数解释	取值范围	默认值
1	timeout	Redis、Memcached	Integer	客户端与服务端连接空闲超时断开时间，参数设为0表示连接永不断开。	0~7200，单位：秒	0
2	maxmemory-policy	Redis、Memcached	Enum	内存使用达到上限时对缓存数据管理策略。 参数说明见 逐出策略 。	- volatile-lru - allkeys-lru - volatile-random - allkeys-random - volatile-ttl - noeviction - volatile-lfu - allkeys-lfu	volatile-lru 说明 如果是2020年7月之前创建的Redis实例，且没有修改过该参数，则默认值为noeviction。如果是2020年7月之后创建的实例，默认值都为volatile-lru。
3	hash-max-ziplist-entries	Redis	Integer	当hash表中记录数少于参数值，使用ziplist编码格式，节约内存。	1~10000	512
4	hash-max-ziplist-value	Redis	Integer	当hash表中各字段长度的最大值小于参数值时，使用ziplist编码格式，节约内存。	1~10000	64
5	list-max-ziplist-entries	Redis	Integer	当列表中记录数少于参数值，使用ziplist编码格式，节约内存。	1~10000	512
6	list-max-ziplist-value	Redis	Integer	当列表中各字段长度的最大值小于参数值时，使用ziplist编码格式，节约内存。	1~10000	64

参数 ID	参数名称	适用引擎	参数类型	参数解释	取值范围	默认值
7	set-max-intset-entries	Redis	Integer	当一个集合仅包含字符串且整型元素数量少于参数值时，使用intset编码格式，节约内存。	1~10000	512
8	zset-max-ziplist-entries	Redis	Integer	当有序集合中记录数少于参数值，使用ziplist编码格式，节约内存。	1~10000	128
9	zset-max-ziplist-value	Redis	Integer	当有序集合中各字段长度的最大值小于参数值时，使用ziplist编码格式，节约内存。	1~10000	64
10	latency-monitor-threshold	Redis	Integer	延时监控的采样时间阈值（最小值）。 当阈值设置为0时，不做监控，也不采样；当阈值设置为大于0时，将监控并记录执行耗时大于阈值的操作。 用户可以通过LATENCY等命令获取统计数据 and 配置、执行采样监控。	0~8640000，单位：毫秒。	0
12	reserved-memory	Redis	Integer	预留内存，只有单机、主备实例有此配置。 预留内存将被分配给实例VM，保证DCS实例后台进程在做持久化和主从同步等任务时拥有足够的内存。 预留内存大小可以调高或者调低，但不能超过实例当前剩余的可用内存，以及最大可用内存的80%。其中实例最大可用内存可参考《分布式缓存服务产品介绍》文档。	实例最大可用内存数的0-80%，同时不能超过实例当前剩余的可用内存，单位：MB。	0

参数 ID	参数名称	适用引擎	参数类型	参数解释	取值范围	默认值
13	notify-keyspace-events	Redis	String	键空间通知，配置该参数后客户端可以通过Redis的订阅与发布功能，来接收那些以某种方式改动了Redis数据集的事件。	当配置为空字符串""或null时，配置会使用默认值Ex。 您可配置为以下字符的任意组合，指定了服务器该发送哪些类型的通知： K：键空间通知，所有通知以__keyspace@__为前缀。 E：键事件通知，所有通知以__keyevent@__为前缀。 g：DEL、EXPIRE、RENAME等类型无关的通用命令的通知。 \$：字符串命令的通知。 l：列表命令的通知。 s：集合命令的通知。 h：哈希命令的通知。	Ex

参数 ID	参数名称	适用引擎	参数类型	参数解释	取值范围	默认值
					z: 有序集合命令的通知。 x: 过期事件: 每当有过期键被删除时发送。 e: 驱逐 (evict) 事件: 每当有键因为 maxmemory 政策而被删除时发送。 其他描述, 请参考表格下方的说明。	

 说明

notify-keyspace-events参数的说明如下:

- 有效值为[K|E|KE][A|g|l|s|h|z|x|e|\$], 即输入的参数中至少要有一个K或者E。
- A为 “g\$lshzxe” 所有参数的集合别名。A与 “g\$lshzxe” 中任意一个不能同时出现。
- 例如, 如果只想订阅键空间中和列表相关的通知, 那么参数就应该设为Kl。若将参数设为字符串"AKE"表示发送所有类型的通知。

响应样例

```
{
  "status": "RUNNING",
  "instance_id": "c08fdc6e-5c25-4185-ab57-c0a5529b727f",
  "redis_config": [
    {
      "description": "How Redis will select what to remove when maxmemory is reached, You can select among five behaviors: volatile-lru : remove the key with an expire set using an LRU algorithm allkeys-lru : remove any key according to the LRU algorithm volatile-random: remove a random key with an expire set allkeys-random: remove a random key, any key volatile-ttl : remove the key with the nearest expire time (minor TTL) noeviction : don't expire at all, just return an error on write operations",
      "param_id": 2,
      "param_name": "maxmemory-policy",
      "param_value": "noeviction",
      "default_value": "noeviction",
      "value_type": "Enum",
      "value_range": "volatile-lru,allkeys-lru,volatile-random,allkeys-random,volatile-ttl,noeviction"
    },
    {
      "description": "Hashes are encoded using a memory efficient data structure when they have a small number of entries",
      "param_id": 3,
```

```

        "param_name": "hash-max-ziplist-entries",
        "param_value": "512",
        "default_value": "512",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Hashes are encoded using a memory efficient data structure when the biggest entry
does not exceed a given threshold",
        "param_id": 4,
        "param_name": "hash-max-ziplist-value",
        "param_value": "64",
        "default_value": "64",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Lists are encoded using a memory efficient data structure when they have a small
number of entries",
        "param_id": 5,
        "param_name": "list-max-ziplist-entries",
        "param_value": "512",
        "default_value": "512",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Lists are encoded using a memory efficient data structure when the biggest entry
does not exceed a given threshold",
        "param_id": 6,
        "param_name": "list-max-ziplist-value",
        "param_value": "64",
        "default_value": "64",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "When a set is composed of just strings that happen to be integers in radix 10 in the
range of 64 bit signed integers.",
        "param_id": 7,
        "param_name": "set-max-intset-entries",
        "param_value": "512",
        "default_value": "512",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Sorted sets are encoded using a memory efficient data structure when they have a
small number of entries",
        "param_id": 8,
        "param_name": "zset-max-ziplist-entries",
        "param_value": "128",
        "default_value": "128",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Sorted sets are encoded using a memory efficient data structure when the biggest
entry does not exceed a given threshold",
        "param_id": 9,
        "param_name": "zset-max-ziplist-value",
        "param_value": "64",
        "default_value": "64",
        "value_type": "Integer",
        "value_range": "1-10000"
    },
    {
        "description": "Close the connection after a client is idle for N seconds (0 to disable)",
        "param_id": 1,

```

```
    "param_name": "timeout",
    "param_value": "0",
    "default_value": "0",
    "value_type": "Integer",
    "value_range": "0-7200"
  },
  {
    "description": "Only events that run in more time than the configured latency-monitor-threshold will be logged as latency spikes. If latency-monitor-threshold is set to 0, latency monitoring is disabled. If latency-monitor-threshold is set to a value greater than 0, all events blocking the server for a time equal to or greater than the configured latency-monitor-threshold will be logged.",
    "param_id": 10,
    "param_name": "latency-monitor-threshold",
    "param_value": "0",
    "default_value": "0",
    "value_type": "Integer",
    "value_range": "0-86400000"
  },
  {
    "description": "The total memory, in bytes, reserved for non-data usage.",
    "param_id": 12,
    "param_name": "reserved-memory",
    "param_value": "0",
    "default_value": "0",
    "value_type": "Integer",
    "value_range": "0-6553"
  },
  {
    "description": "Redis can notify Pub or Sub clients about events happening in the key space",
    "param_id": 13,
    "param_name": "notify-keyspace-events",
    "param_value": null,
    "default_value": null,
    "value_type": "regular",
    "value_range": "([KE]+([A])([g$!shzxe]+)){0,11}"
  }
],
"config_status": "SUCCESS",
"config_time": ""
}
```

状态码

操作成功的状态码如表8-58所示，其他响应见表7-1。

表 8-58 状态码

状态码	描述
200	查询实例参数成功。

8.4 备份和恢复

8.4.1 备份指定实例

功能介绍

备份指定的缓存实例。

 说明

只有主备和集群类型的缓存实例支持备份恢复操作，单机实例不支持备份恢复操作。

URI

POST /v1.0/{project_id}/instances/{instance_id}/backups

参数说明见[表8-59](#)。

表 8-59 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

参数说明见[表8-60](#)。

表 8-60 参数说明

参数	类型	必选	说明
remark	String	否	备份缓存实例的备注信息。

请求示例

POST https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/backups

```
{
  "remark": "Backup instances"
}
```

响应消息

响应参数

参数说明见[表8-61](#)。

表 8-61 参数说明

参数	类型	说明
backup_id	String	备份记录ID。

响应示例

```
{
  "backup_id": "548ceeff-2cbb-47ab-9a1c-7b085a8c08d7"
}
```

状态码

操作成功的状态码如表8-62所示，其他响应见表7-1。

表 8-62 状态码

状态码	描述
200	备份指定缓存实例成功。

8.4.2 恢复指定实例

功能介绍

恢复指定的缓存实例。

说明

只有主备和集群类型的缓存实例支持备份恢复操作，单机实例不支持备份恢复操作。

URI

POST /v1.0/{project_id}/instances/{instance_id}/restores

参数说明见表8-63。

表 8-63 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。

请求消息

请求参数

参数说明见表8-64。

表 8-64 参数说明

参数	类型	必选	说明
remark	String	否	恢复缓存实例的备注信息。
backup_id	String	是	备份ID。

请求示例

```
POST https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/restores

{
  "remark": "restore instance",
  "backup_id": "8ba256cb-e5ac-44f6-a3da-c03d8f0e5029"
}
```

响应消息

响应参数

参数说明见表8-65。

表 8-65 参数说明

参数	类型	说明
restore_id	String	恢复记录ID。

响应示例

```
{
  "restore_id": "a6155972-800c-4170-a479-3231e907d2f6"
}
```

状态码

操作成功的状态码见表8-66所示，其他响应见表7-1。

表 8-66 状态码

状态码	描述
200	恢复缓存实例成功。

8.4.3 查询实例备份信息

功能介绍

查询指定缓存实例的备份信息列表。

URI

```
GET /v1.0/{project_id}/instances/{instance_id}/backups?
start={start}&limit={limit}&beginTime={beginTime}&endTime={endTime}
```

参数说明见表8-67。

表 8-67 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。
start	Integer	否	待查询的备份记录的起始序号，默认起始序号为1。
limit	Integer	否	每页显示条数，最小值为1，若不设置该参数，默认显示10条。
beginTime	String	否	查询开始时间。格式：yyyyMMddHHmmss，如：20170718235959。
endTime	String	否	查询结束时间。格式：yyyyMMddHHmmss，如：20170718235959。

请求消息

请求参数

无

请求示例

```
GET https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/backups?
start={start}&limit={limit}&beginTime={beginTime}&endTime={endTime}
```

响应消息

响应参数

参数说明见[表8-68](#)。

表 8-68 参数说明

参数	类型	说明
backup_record_response	Array	备份信息的详情数组，具体请参考 表8-69 。
total_num	Integer	返回记录数。

表 8-69 backup_record_response 参数说明

参数	类型	说明
status	String	备份状态。 • waiting：等待中。 • backuping：备份中。 • succeed：备份成功。 • failed：备份失败。 • expired：备份文件过期。 • deleted：已手动删除备份文件。
remark	String	备份缓存实例的备注信息。
period	String	备份执行时间段。
progress	String	备份进度。
size	Long	备份文件大小（Byte）。
instance_id	String	实例ID。
backup_id	String	备份ID。
created_at	String	备份任务创建时间。
updated_at	String	备份完成时间。
execution_at	String	备份任务执行时间。
backup_type	String	备份类型。 • manual：表示备份类型为手动备份 • auto：表示备份类型为自动备份
backup_name	String	备份记录名称。
error_code	String	备份失败后错误码，参见 表8-70 。
is_support_restore	String	是否可以进行恢复操作，取值为TRUE或FALSE。

表 8-70 DCS 备份恢复错误码

错误码	说明
dcx.08.0001	启动备份恢复工具失败。
dcx.08.0002	执行超时。
dcx.08.0003	删除桶失败。
dcx.08.0004	获取ak/sk失败。
dcx.08.0005	创建桶失败。

错误码	说明
dcs.08.0006	查询备份数据大小失败。
dcs.08.0007	恢复时同步数据失败。
dcs.08.0008	自动备份任务未运行，实例正在运行其他任务。

响应示例

```
{
  "backup_record_response": [
    {
      "status": "succeed",
      "remark": "001",
      "period": null,
      "progress": "100.00",
      "size": 880232,
      "instance_id": "5560df16-cebf-4473-95c4-d1b573c16e79",
      "backup_id": "4631832a-14c6-45b0-a0b3-3abd8f591ad1",
      "created_at": "2019-05-10T08:31:16.166Z",
      "updated_at": "2019-05-10T08:32:30.546Z",
      "execution_at": "2019-05-10T08:31:21.461Z",
      "backup_type": "manual",
      "backup_name": "backup_20190510163116",
      "error_code": null,
      "is_support_restore": "TRUE"
    }
  ],
  "total_num": 1
}
```

状态码

操作成功的状态码如[表8-71](#)所示，其他响应见[表7-1](#)。

表 8-71 状态码

状态码	描述
200	查询缓存实例备份信息成功。

8.4.4 查询实例恢复记录

功能介绍

查询指定缓存实例的恢复记录列表。

URI

GET /v1.0/{project_id}/instances/{instance_id}/restores?
start={start}&limit={limit}&beginTime={beginTime}&endTime={endTime}

参数说明见[表8-72](#)。

表 8-72 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。
start	Integer	否	待查询的恢复记录的起始序号，默认起始序号为1。
limit	Integer	否	每页显示条数，最小值为1，若不设置该参数，默认显示10条。
beginTime	String	否	查询开始时间。格式：yyyyMMddHHmmss，如：20170718235959。
endTime	String	否	查询结束时间。格式：yyyyMMddHHmmss，如：20170718235959。

请求消息

请求参数

无

请求示例

```
GET https://{dcs_endpoint}/v1.0/{project_id}/instances/{instance_id}/restores?
start={start}&limit={limit}&beginTime={beginTime}&endTime={endTime}
```

响应消息

响应参数

参数说明见[表8-73](#)。

表 8-73 参数说明

参数	类型	说明
restore_record_response	Array	实例恢复记录的详情数组。
total_num	Integer	返回记录数。

表 8-74 restore_record_response 参数说明

参数	类型	说明
status	String	恢复状态。 • waiting：等待中 • restoring：恢复中 • succeed：恢复成功 • failed：恢复失败
progress	String	恢复进度。
restore_id	String	恢复记录ID。
backup_id	String	备份ID。
restore_remark	String	恢复备注信息。
backup_remark	String	备份备注信息。
created_at	String	恢复任务创建时间。
updated_at	String	恢复完成时间。
restore_name	String	恢复记录名称
backup_name	String	备份记录名称。
error_code	String	恢复失败后错误码，参见 表8-70 。

响应示例

```
{
  "restore_record_response": [
    {
      "status": "succeed",
      "progress": "100.00",
      "restore_id": "a6155972-800c-4170-a479-3231e907d2f6",
      "backup_id": "f4823e9e-fe9b-4ffd-be79-4e5d6de272bb",
      "restore_remark": "doctest",
      "backup_remark": null,
      "created_at": "2017-07-18T21:41:20.721Z",
      "updated_at": "2017-07-18T21:41:35.182Z",
      "restore_name": "restore_20170718214120",
      "backup_name": "backup_20170718000002",
      "error_code": null
    }
  ],
  "total_num": 1
}
```

状态码

操作成功的状态码如[表8-75](#)所示，其他响应见[表7-1](#)。

表 8-75 状态码

状态码	描述
200	查询实例恢复记录成功。

8.4.5 删除备份文件

功能介绍

删除缓存实例已备份的文件。

URI

DELETE /v1.0/{project_id}/instances/{instance_id}/backups/{backup_id}

参数说明见[表8-76](#)。

表 8-76 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。
instance_id	String	是	实例ID。
backup_id	String	是	备份记录ID

示例

DELETE https://{*dcg_endpoint*}/v1.0/885cacf2d49d4bb6931ae668e9c07553/instances/e016385d-b9fa-4bf0-9f38-9379f4a5293f/backups/75509c85-50a6-4525-ad56-a1bb62e84570

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-77](#)。

表 8-77 参数说明

参数	类型	说明
message	String	删除备份文件结果说明。

响应示例

```
{
  "message": ""
}
```

状态码

操作成功的状态码如[表8-78](#)所示，其他响应见[表7-1](#)。

表 8-78 状态码

状态码	描述
200	删除备份文件成功。

8.4.6 获取备份文件下载链接

功能介绍

获取指定实例的备份文件下载链接，用于下载备份文件。

 说明

该接口当前仅在华北-北京四区域开放。

URI

POST /v1.0/{project_id}/instances/{instance_id}/backups/{backup_id}/links

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
instance_id	是	String	实例ID。
backup_id	是	String	备份记录ID。具体请参考 查询实例备份信息 接口，获取备份ID。

请求消息

请求参数

参数	是否必选	参数类型	描述
expiration	是	String	设置URL的有效期，必须在5分钟和24小时之内，单位为秒。

请求示例

```
{
  "expiration":300
}
```

响应消息

状态码为 200 时，响应参数和响应示例如下：

响应参数

参数	参数类型	描述
bucket_name	String	OBS桶名。
file_path	String	OBS桶内文件路径。
links	Array	备份文件下载链接集合，链接数最大为64个。

参数	参数类型	描述
file_name	String	备份文件名称。
link	String	备份文件下载链接地址。

响应示例

```
{
  "bucket_name": "bucket6d1b61f3d7434f60b7e465f4037d73f3",
  "file_path": "3dc6ba15-ba97-4624-bcbd-3cc08d6336b0/20190924204307_redis.rdb",
  "links": [
    {
      "file_name": "20190924204307_redis.rdb",
      "link": "https://bucket6d1b61f3d7434f60b7e465f4037d73f3.obs.cn-north-4.myhuaweicloud.com:443/3dc6ba15-ba97-4624-bcbd-3cc08d6336b0/20190924204307_redis.rdb?AWSAccessKeyId=IMRORRY3GETV9CEH3GE0&Expires=1569329601&Signature=NGhdEGwHImXNyYryx3AYqfAvoA%3D"
    }
  ]
}
```

返回值

返回值	描述
200	获取备份文件下载链接成功。

8.5 标签管理

8.5.1 查询租户所有标签

功能介绍

查询租户在指定Project中实例类型的所有资源标签集合。

 说明

该接口当前仅在华北-北京四区域开放。

URI

GET /v1.0/{project_id}/dcs/tags

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。

请求参数

无

响应参数

状态码为 200 时:

参数	参数类型	描述
key	String	标签键。 • 最大长度36个unicode字符。
values	Array of strings	标签值。 • 每个值最大长度43个unicode字符。 • 可以为空字符串。

响应示例


```
{
  "tags": [
    {
      "key": "001",
      "values": [
        "002"
      ]
    },
    {
      "key": "003",
      "values": [
        "004"
      ]
    }
  ]
}
```

返回值

返回值	描述
200	查询租户所有标签成功。

错误码

无

8.5.2 批量添加或删除标签

功能介绍

为指定实例批量添加标签，或批量删除标签。

 说明

该接口当前仅在华北-北京四区域开放。

URI

POST /v1.0/{project_id}/dcs/{instance_id}/tags/action

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
instance_id	是	String	实例ID。

URI示例:

```
Post https://{dcs_endpoint}/v1.0/{project_id}/dcs/{instance_id}/tags/action
```

请求参数

请求参数

表 8-79 请求参数

参数	是否必选	参数类型	描述
action	是	String	操作标识：仅限于create（创建）、delete（删除）。
tags	是	Array of objects	标签列表。具体参数，请参考表8-80。

表 8-80 tags 参数说明

参数	是否必选	参数类型	描述
key	是	String	标签键。 - 不能为空。 - 对于同一资源键值唯一。 - 长度不超过36个字符。
value	否	String	标签值。 - action为create时，必选。action为delete时，非必选。 - 长度不超过43个字符。

请求实例

```
{
  "action": "create",
  "tags": [
    {
      "key": "dcs001",
      "value": "002"
    },
    {
      "key": "dcs003",
      "value": "004"
    }
  ]
}
```

响应参数

无

返回值

返回值	描述
200	创建/删除标签成功。

错误码

无

8.5.3 查询单个实例标签

功能介绍

通过实例ID查询该实例所有标签。

 说明

该接口当前仅在华北-北京四区域开放。

URI

GET /v1.0/{project_id}/instances/{instance_id}/tags

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码为 200 时:

表 8-81 响应参数

参数	参数类型	描述
tags	Array of objects	标签列表。具体参数请参见 表8-82

表 8-82 tags 参数说明

参数	参数类型	描述
key	String	标签键。 - 不能为空。 - 对于同一资源键值唯一。 - 长度不超过36个字符。

参数	参数类型	描述
value	String	标签值。 - 长度不超过43个字符。 - 可以为空字符串。

响应示例

状态码为 200 时:

```
{
  "tags": [
    {
      "key": "001",
      "value": "002"
    },
    {
      "key": "003",
      "value": "004"
    }
  ]
}
```

返回值

返回值	描述
200	查询实例标签成功。

错误码

无

8.6 其他接口

8.6.1 查询产品规格列表

功能介绍

在创建缓存实例时，需要配置订购的产品ID（即product_id），可通过该接口查询产品规格。例如，要订购按需计费、实例类型为Redis3.0的主备实例，可从接口响应消息中，查找charging_type字段为Hourly、spec_code字段为dcs.master_standby的记录对应的product_id，该字段的值即是创建缓存实例时需要配置的产品ID。

URI

GET /v1.0/products

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-83](#)。

表 8-83 参数说明

参数	类型	说明
products	Array	产品列表。

表 8-84 products 参数说明

参数	类型	备注
product_id	String	DCS服务对外销售的产品ID。 参数取值格式为：spec_code+ “-” + 计费模式缩写，计费模式缩写取值范围有：h、m、y。其中，h表示计费模式为按需；m表示计费模式为包月；y表示计费模式为包年。 例如：redis.single.au1.large.1-h，表示规格redis.single.au1.large.1的按需计费产品ID。 说明 在查询产品规格中获取product_id时，请务必确保获取的product_id对应的计费模式和您选择的计费模式一致。例如，要订购按需计费、实例类型为Redis3.0的主备实例，可从接口响应消息中，查找charging_type字段为Hourly、spec_code字段为dcs.master_standby的记录对应的product_id。
spec_code	String	DCS的产品规格编码。 • dcs.single_node • dcs.master_standby • dcs.cluster

参数	类型	备注
cache_mode	String	缓存实例类型。取值范围如下： • single：表示单机 • ha：表示主备实例 • cluster：表示集群 • proxy：表示Proxy集群 • ha_rw_split：表示读写分离
product_type	String	Redis缓存实例的产品类型。
cpu_type	String	CPU架构类型。
storage_type	String	存储类型。
details	Array	规格详细描述，具体参数，请参考 表8-85 。
engine	String	缓存引擎类型。
engine_versions	String	支持的引擎版本号。
spec_details	String	产品的规格大小。具体以查询结果为准。
spec_details2	String	DCS的规格详细信息，如最大连接数和最大使用内存等。
charging_type	String	计费类型，取值为：Yearly、Monthly、Hourly。
price	doubule	价格。（已废弃，无实际意义，请不要使用。）
currency	String	货币单位。
prod_type	String	产品类型，取值为instance或者obs_space。
cloud_service_type_code	String	云服务类型编码。
cloud_resource_type_code	String	云资源类型编码。
flavors	Array	有资源的可用区，具体参数，请参考 表8-86 。
billing_factor	String	计费因子。

表 8-85 details 参数说明

参数	类型	备注
capacity	String	缓存实例的内存规格。
max_bandwidth	String	该规格支持的最大带宽。
max_clients	String	该规格支持的最大客户端数，一般等于最大连接数。

参数	类型	备注
max_connections	String	该规格支持的最大连接数。
max_in_bandwidth	String	该规格支持的最大接入带宽，一般等于最大带宽。
max_memory	String	实际可使用的最大内存。
tenant_ip_count	String	该规格实例对应的租户侧IP数量。
sharding_num	String	该规格实例支持的分片数。
proxy_num	String	该规格Proxy实例支持的Proxy节点数量。如果不是Proxy实例，该参数为0。
db_number	String	该规格实例的DB数量。

表 8-86 flavors 参数说明

参数	类型	备注
capacity	String	缓存实例的内存规格。
unit	String	容量单位
available_zones	Array	可用区ID。

响应示例

```
{
  "product_id": "dcs.master_standby-h",
  "spec_code": "dcs.master_standby",
  "cache_mode": "ha",
  "product_type": "generic",
  "cpu_type": "x86_64",
  "storage_type": "DRAM",
  "details": {
    "capacity": 2,
    "max_memory": 1.5,
    "max_connections": 10000,
    "max_clients": 5000,
    "max_bandwidth": 512,
    "max_in_bandwidth": 42,
    "tenant_ip_count": 3,
    "sharding_num": 1,
    "proxy_num": 0,
    "db_number": 256
  },
  "engine": "redis",
  "engine_versions": "3.0",
  "spec_details": "[{\\"mem\\":\\"2,4,8,16,32,64\\"}]",
  "spec_details2": "[{\\"capacity\\":2,\\"max_memory\\":1.5,\\"max_connections\\":10000,\\"max_clients\\":5000,\\"max_bandwidth\\":512,\\"max_in_bandwidth\\":42,\\"tenant_ip_count\\":3,\\"sharding_num\\":1,\\"proxy_num\\":0,\\"db_number\\":256},{\\"capacity\\":4,\\"max_memory\\":3.2,\\"max_connections\\":10000,\\"max_clients\\":5000,\\"max_bandwidth\\":1536,\\"max_in_bandwidth\\":64,\\"tenant_ip_count\\":3,\\"sharding_num\\":1,\\"proxy_num\\":0,\\"db_number\\":256},{\\"capacity\\":8,\\"max_memory\\":6.4,\\"max_connections\\":10000,\\"max_clients\\":5000,\\"max_bandwidth\\":1536,\\"max_in_bandwidth\\":64,\\"tenant_ip_count\\":3,\\"sharding_num\\":1,\\"proxy_num\\":0,\\"db_number\\":256},{\\"capacity\\":16,\\"max_memory\\":12.8,\\"max_connections\\":10000,\\"max_clients\\":5000,\\"max_bandwidth\\":1536,\\"max_in_bandwidth\\":64,\\"tenant_ip_count\\":3,\\"sharding_num\\":1,\\"proxy_num\\":0,\\"db_number\\":256}]"
```

```
\":16,\"max_memory\":12.8,\"max_connections\":10000,\"max_clients\":5000,\"max_bandwidth\n\":3072,\"max_in_bandwidth\":85,\"tenant_ip_count\":3,\"sharding_num\":1,\"proxy_num\":0,\"db_number\n\":256},{\"capacity\":32,\"max_memory\":25.6,\"max_connections\":10000,\"max_clients\n\":5000,\"max_bandwidth\":3072,\"max_in_bandwidth\":85,\"tenant_ip_count\":3,\"sharding_num\n\":1,\"proxy_num\":0,\"db_number\":256},{\"capacity\":64,\"max_memory\":51.2,\"max_connections\n\":12000,\"max_clients\":5000,\"max_bandwidth\":5120,\"max_in_bandwidth\":128,\"tenant_ip_count\n\":3,\"sharding_num\":1,\"proxy_num\":0,\"db_number\":256}],\n  \"charging_type\": \"Hourly\",\n  \"price\": 0.0,\n  \"currency\": \"\",\n  \"prod_type\": \"instance\",\n  \"cloud_service_type_code\": \"hws.service.type.dcs\",\n  \"cloud_resource_type_code\": \"hws.resource.type.dcs\",\n  \"flavors\": [{\n    \"capacity\": \"2\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  },\n  {\n    \"capacity\": \"4\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  },\n  {\n    \"capacity\": \"8\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  },\n  {\n    \"capacity\": \"16\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  },\n  {\n    \"capacity\": \"32\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  },\n  {\n    \"capacity\": \"64\",\n    \"unit\": \"GB\",\n    \"available_zones\": [\"882f6e449e3245dbb8c1c0fafa494c89\",\n      \"ae04cf9d61544df3806a3feeb401b204\",\n      \"d573142f24894ef3bd3664de068b44b0\"]\n  }\n],\n  \"billing_factor\": \"Duration\"\n}
```

状态码

操作成功的状态码如[表8-87](#)所示，其他响应见[表7-1](#)。

表 8-87 状态码

状态码	描述
200	查询规格列表成功。

8.6.2 查询租户的试用权限

功能介绍

查询租户可以参与的免费使用权限。

 说明

该接口仅“华北-北京一”、“华东-上海二”和“华南-广州” region支持，其他region暂未支持。

URI

GET /v1.0/{domain_id}/trials

参数说明见[表8-88](#)。

表 8-88 参数说明

参数	类型	必选	说明
domain_id	String	是	租户ID。

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-89](#)。

表 8-89 参数说明

参数	类型	说明
domain_id	String	租户ID。

参数	类型	说明
domain_name	String	租户名称。
free_trials	JSON	免费试用信息。
one_yuan_trials	JSON	1元试用信息。

表 8-90 free_trials 和 one_yuan_trials 中的参数说明

参数	类型	说明
engine	String	引擎类型。取值包括Redis和Memcached。
version	String	引擎版本。
support	Boolean	是否支持试用。
cause_id	String	不支持的原因ID。
message	String	额外消息。

表 8-91 cause_id 取值说明

cause_id	说明
111400129	暂时没有免费试用活动。
111400125	用户免费试用限额已经用完，或者暂不支持参加该免费试用活动。

响应示例

```
{
  "domain_id": "xx",
  "domain_name": "xx",
  "free_trials": [{
    "engine": "redis",
    "support": true,
    "cause_id": "xxx",
    "message": "xx"
  }]
}
```

状态码

操作成功的状态码如[表8-92](#)所示，其他响应见[表7-1](#)。

表 8-92 状态码

状态码	描述
200	查询租户试用权限成功。

8.6.3 查询租户配额

功能介绍

查询租户默认可以创建的实例数和总内存的配额限制，以及可以申请配额的最大值和最小值。不同的租户在不同的区域配额可能不同。

URI

GET /v1.0/{project_id}/quota

参数说明见[表8-93](#)。

表 8-93 参数说明

参数	类型	必选	说明
project_id	String	是	项目ID。

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-94](#)。

表 8-94 参数说明

参数	类型	必选	说明
quotas	JSON	是	配额信息。参见 表8-95 。

表 8-95 quotas 参数说明

参数	类型	必选	说明
resources	Array	是	配额列表。参见表8-96。
resource_user	JSON	是	资源租户信息。参见表8-97。

表 8-96 resources 参数说明

参数	类型	说明
quota	Integer	可以创建的实例最大数和总内存的配额限制。
used	Integer	已创建的实例个数和已使用的内存配额。
type	String	支持instance、ram两种。 - instance表示实例配额。 - ram表示内存配额。
unit	String	资源的计数单位。 - 当type为instance时，无单位。 - 当type为ram时，单位为GB。
max	Integer	- 当type为instance时，表示可申请实例配额的最大值。 - 当type为ram时，表示可申请内存配额的最大值。
min	Integer	- 当type为instance时，表示可申请实例配额的最小值。 - 当type为ram时，表示可申请内存配额的最小值。

表 8-97 resource_user 参数说明

参数	类型	说明
tenant_id	String	资源租户id。
tenant_name	Integer	资源租户名称。

响应示例

```
{
  "quotas": {
```

```
    "resources": [
      {
        "quota": 10,
        "used": 3,
        "type": "instance",
        "min": 1,
        "max": 10,
        "unit": null
      },
      {
        "quota": 800,
        "used": 22,
        "type": "ram",
        "min": 1,
        "max": 800,
        "unit": "GB"
      }
    ],
    "resource_user": {
      "tenant_id": "836152f9838a44089f40f3cf6fd432bf",
      "tenant_name": "op_svc_dcs_003"
    }
  }
}
```

状态码

操作成功的状态码如表8-98所示，其他响应见表7-1。

表 8-98 状态码

状态码	描述
200	查询租户配额成功。

8.6.4 查询维护时间窗时间段

功能介绍

查询维护时间窗开始时间和结束时间。

URI

GET /v1.0/instances/maintain-windows

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-99](#)。

表 8-99 响应参数说明

参数	类型	备注
maintain_windows	Array	支持的维护时间窗列表

表 8-100 maintain_windows 参数说明

参数	类型	备注
seq	Integer	序号。
begin	String	维护时间窗开始时间。
end	String	维护时间窗结束时间。
default	Boolean	是否为默认时间段。

响应示例

```
{
  "maintain_windows": [
    {
      "seq": 1,
      "begin": "22",
      "end": "02",
      "default": false
    },
    {
      "seq": 2,
      "begin": "02",
      "end": "06",
      "default": true
    },
    {
      "seq": 3,
      "begin": "06",
      "end": "10",
      "default": false
    },
    {
      "seq": 4,
      "begin": "10",
      "end": "14",
      "default": false
    },
    {
      "seq": 5,
      "begin": "14",
      "end": "18",
      "default": false
    },
    {
      "seq": 6,
      "begin": "18",
      "end": "22",
      "default": false
    }
  ]
}
```

```
    ]  
  }
```

状态码

操作成功的状态码如[表8-101](#)所示，其他响应见[表7-1](#)。

表 8-101 状态码

状态码	描述
200	查询维护时间窗时间段成功。

8.6.5 查询可用区信息

功能介绍

在创建缓存实例时，需要配置实例所在的可用区ID，可通过该接口查询可用区的ID。例如，在华北-北京1区域有可用资源的分区创建缓存实例，可从接口响应消息中，查找resource_availability字段为true的可用区，该记录对应的id即是创建缓存实例时需要配置的可用区ID。

URI

GET /v1.0/availableZones

请求消息

请求参数

无。

请求示例

无。

响应消息

响应参数

参数说明见[表8-102](#)。

表 8-102 参数说明

参数	类型	说明
regionId	String	区域ID。
available_zones	Array	可用区数组，具体请参考 表8-103 。

表 8-103 available_zones 参数说明

参数	类型	说明
id	String	可用区ID。
code	String	可用区编码。
name	String	可用区名称。
port	String	可用区端口号。
resource_availability	String	分区上是否还有可用资源。 • true：还有资源。 • false：资源已售罄。

响应示例

```
{
  "regionId": "XXXXXX",
  "available_zones": [
    {
      "id": "f84448fd537f46078dd8bd776747f573",
      "code": "XXXXXX",
      "name": "XXXXXX",
      "port": "8003",
      "resource_availability": "true"
    },
    {
      "id": "12c47a78666b4e438cd0c692b9860387",
      "code": "XXXXXX",
      "name": "XXXXXX",
      "port": "8002",
      "resource_availability": "true"
    },
    {
      "id": "0725858e0d26434f9aa3dc5fc40d5697",
      "code": "XXXXXX",
      "name": "XXXXXX",
      "port": "8009",
      "resource_availability": "true"
    }
  ]
}
```

状态码

操作成功的状态码如[表8-104](#)所示，其他响应见[表7-1](#)。

表 8-104 状态码

状态码	描述
200	查询成功。

8.7 实例详情

8.7.1 变更指定实例的带宽

功能介绍

Redis 4.0及之后版本的实例，支持临时扩容缓存实例带宽，用于临时解决业务流量高峰，带宽超限的问题。实例带宽扩容后，带宽为原带宽的2倍。Proxy集群实例带宽上限为10000Mbit/s，达到上限后无法扩容。一次带宽临时扩容的有效期为7天，到期后恢复为原带宽，失效1天后可再次申请扩容，一个实例最多扩容3次。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/instances/{instance_id}/bandwidth

表 8-105 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法请参见 获取项目ID 。
instance_id	是	String	实例ID。

请求参数

无

响应参数

状态码： 400

表 8-106 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。 最大长度： 1024
error_code	String	错误码。 最大长度： 9
error_ext_msg	String	扩展错误信息（ 暂未使用，赋值为null ）。 最大长度： 1024

状态码： 500

表 8-107 响应 Body 参数

参数	参数类型	描述
error_msg	String	错误信息。 最大长度：1024
error_code	String	错误码。 最大长度：9
error_ext_msg	String	扩展错误信息（暂未使用，赋值为null）。 最大长度：1024

请求示例

```
PUT https://{dcs_endpoint}/v2/{project_id}/instances/{instance_id}/bandwidth
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dcs.v2.region.DcsRegion;
import com.huaweicloud.sdk.dcs.v2.*;
import com.huaweicloud.sdk.dcs.v2.model.*;

public class UpdateInstanceBandwidthSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        DcsClient client = DcsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DcsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceBandwidthRequest request = new UpdateInstanceBandwidthRequest();
        try {
```

```
        UpdateInstanceBandwidthResponse response = client.updateInstanceBandwidth(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdcs.v2.region.dcs_region import DcsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdcs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = DcsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DcsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceBandwidthRequest()
        response = client.update_instance_bandwidth(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dcs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dcs/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := dcs.NewDcsClient(
    dcs.DcsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateInstanceBandwidthRequest{}
response, err := client.UpdateInstanceBandwidth(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	带宽变更成功。
400	非法请求。
500	内部服务错误。

错误码

请参见[错误码](#)。