

数据仓库服务

API 参考

文档版本 01
发布日期 2025-01-06



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。

目录

1 使用前必读	1
2 API 概述	3
3 如何调用 API	6
3.1 构造请求	6
3.2 认证鉴权	10
3.3 返回结果	10
4 快速入门	13
5 API 说明	19
5.1 可用区	19
5.1.1 查询可用区列表	19
5.2 集群管理	23
5.2.1 创建集群	23
5.2.2 V2 创建集群	30
5.2.3 创建逻辑集群	38
5.2.4 物理集群转换到逻辑集群	45
5.2.5 查询集群列表	48
5.2.6 查询逻辑集群列表	59
5.2.7 查询逻辑集群任务信息	66
5.2.8 查询逻辑集群磁盘信息	70
5.2.9 查询节点列表	75
5.2.10 查询逻辑集群可用 ring 环节点信息	81
5.2.11 查询集群拓扑 ring 环节点信息	86
5.2.12 查询集群详情	91
5.2.13 查询节点类型	104
5.2.14 删除集群	110
5.2.15 删除逻辑集群	114
5.2.16 编辑逻辑集群	117
5.2.17 重启逻辑集群	123
5.2.18 切换逻辑集群开关	127
5.2.19 删除空闲节点	130
5.2.20 从空闲节点扩容	134
5.2.21 恢复重分布	140

5.2.22 暂停重分布.....	143
5.2.23 重启集群.....	146
5.2.24 扩容集群.....	149
5.2.25 重置密码.....	153
5.2.26 创建集群前检查.....	157
5.2.27 集群缩容.....	163
5.2.28 下发重分布.....	168
5.2.29 查询重分布详情.....	172
5.2.30 查询专属分布式存储池列表.....	179
5.2.31 查询资源统计信息列表.....	183
5.2.32 主备恢复.....	187
5.2.33 解除只读.....	190
5.2.34 磁盘扩容.....	194
5.2.35 修改运维时间窗.....	198
5.2.36 查询集群 CN 节点.....	202
5.2.37 批量增加 CN 节点.....	206
5.2.38 批量删除 CN 节点.....	210
5.2.39 查询集群参数组.....	214
5.2.40 查询集群参数配置.....	218
5.2.41 修改集群参数配置.....	222
5.2.42 快照统计信息.....	227
5.2.43 查询合适的缩容数.....	231
5.2.44 查询参数修改审计记录.....	234
5.2.45 保存集群描述信息.....	239
5.2.46 查询单个实例.....	243
5.2.47 启动集群.....	250
5.2.48 停止集群.....	253
5.3 快照管理.....	257
5.3.1 创建快照.....	257
5.3.2 查询快照列表.....	262
5.3.3 查询快照策略.....	266
5.3.4 查询快照详情.....	270
5.3.5 删除手动快照.....	276
5.3.6 删除快照策略.....	279
5.3.7 恢复快照到新集群.....	282
5.3.8 用户恢复表名检测.....	288
5.3.9 恢复表.....	293
5.3.10 获取跨区域快照可用 Region.....	298
5.3.11 设置跨区域备份配置.....	302
5.3.12 查询所有跨区域快照配置.....	306
5.3.13 删除跨区域备份配置.....	310
5.3.14 复制快照.....	313

5.3.15 查询集群快照列表.....	317
5.3.16 添加快照策略.....	324
5.4 事件管理.....	328
5.4.1 查询事件列表.....	329
5.4.2 查询事件配置.....	333
5.4.3 查询订阅事件.....	338
5.4.4 创建订阅事件.....	342
5.4.5 更新订阅事件.....	348
5.4.6 删除订阅事件.....	354
5.5 数据源.....	358
5.5.1 查询数据源.....	358
5.5.2 创建数据源.....	362
5.5.3 更新数据源.....	367
5.5.4 删除数据源.....	371
5.6 审计日志.....	374
5.6.1 查询审计日志记录.....	374
5.7 资源管理.....	378
5.7.1 设置资源管理开关.....	378
5.7.2 查询资源管理状态.....	382
5.7.3 查询资源统计.....	386
5.7.4 添加资源池.....	390
5.7.5 删除资源池.....	397
5.7.6 查询资源池列表.....	401
5.7.7 获得资源池详细信息.....	404
5.7.8 更新资源池配置信息.....	409
5.7.9 资源池关联用户.....	415
5.7.10 获得资源池绑定用户列表.....	420
5.7.11 资源池解绑用户.....	424
5.7.12 添加资源管理计划.....	429
5.7.13 查询资源管理计划列表.....	433
5.7.14 启动资源管理计划.....	438
5.7.15 查看计划执行日志.....	442
5.7.16 查询某个资源管理计划详细信息.....	446
5.7.17 停止资源管理计划.....	452
5.7.18 删除资源管理计划.....	455
5.7.19 添加资源管理计划阶段.....	459
5.7.20 查询资源管理计划阶段详细信息.....	466
5.7.21 删除资源管理计划阶段.....	470
5.7.22 切换资源管理计划阶段.....	474
5.7.23 查询集群模式空间信息.....	478
5.7.24 更新模式空间限额.....	483
5.8 告警管理.....	486

5.8.1 查询告警配置.....	486
5.8.2 查询告警详情列表.....	491
5.8.3 查询告警统计列表.....	495
5.8.4 查询告警订阅列表.....	500
5.8.5 创建告警订阅.....	504
5.8.6 更新告警订阅.....	509
5.8.7 删除告警订阅.....	515
5.9 连接管理.....	519
5.9.1 申请域名.....	519
5.9.2 修改集群域名.....	522
5.9.3 删除集群域名.....	526
5.9.4 获取集群可绑定的 ELB 列表.....	530
5.9.5 集群绑定 ELB.....	534
5.9.6 集群解绑 ELB.....	537
5.9.7 集群绑定 EIP.....	541
5.9.8 集群解绑 EIP.....	544
5.10 标签管理.....	547
5.10.1 查询项目标签.....	547
5.10.2 查询集群标签.....	551
5.10.3 批量添加标签.....	555
5.10.4 批量删除标签.....	559
5.11 配额管理.....	563
5.11.1 查询配额.....	563
5.12 容灾管理.....	567
5.12.1 创建容灾.....	567
5.12.2 启动容灾.....	572
5.12.3 查询容灾列表.....	576
5.12.4 查询容灾详情.....	581
5.12.5 查询容灾恢复进度详情.....	587
5.12.6 查询可用容灾集群列表.....	591
5.12.7 更新容灾配置.....	596
5.12.8 灾备切换.....	601
5.12.9 容灾异常切换.....	605
5.12.10 恢复容灾.....	608
5.12.11 停止容灾.....	612
5.12.12 检查容灾名称.....	615
5.12.13 删除容灾.....	619
5.13 任务管理.....	622
5.13.1 查询 job 进度.....	622
5.14 主机监控.....	626
5.14.1 查询主机概览.....	626
5.14.2 查询磁盘信息.....	631

5.14.3 获取网卡状态.....	636
5.14.4 查询性能监控指标.....	641
5.14.5 查询历史监控数据.....	645
5.14.6 查询表倾斜或脏页率信息.....	650
5.14.7 查询 SQL 列表.....	655
5.14.8 查询 SQL 执行信息.....	677
5.14.9 查询集群使用指标列表.....	684
5.14.10 获取指定指标相关采集数据.....	687
5.15 数据库权限管理.....	689
5.15.1 同步 IAM 用户到数据库.....	689
5.15.2 查询所有数据库用户/角色.....	693
5.15.3 查询指定用户信息.....	697
5.15.4 修改指定用户信息.....	701
5.15.5 修改数据库对象权限.....	706
5.15.6 查询数据库对象权限.....	713
5.15.7 获得集群运维账户状态.....	718
5.15.8 执行运维用户操作.....	722
5.16 日志管理.....	731
5.16.1 获取 LTS 日志列表.....	731
5.16.2 开启云服务日志.....	735
5.16.3 关闭云服务日志.....	738
5.17 升级管理.....	741
5.17.1 获取集群升级记录.....	742
5.17.2 获取集群可升级的目标版本.....	746
5.17.3 下发集群升级相关操作.....	750
6 应用示例.....	755
6.1 使用 Postman 调用创建集群接口.....	755
6.2 使用 Postman 调用创建快照接口.....	757
7 权限及授权项说明.....	760
8 附录.....	764
8.1 状态码.....	764
8.2 错误码.....	766
8.3 创建 VPC.....	774
8.4 获取项目 ID.....	774
8.5 获取账号 ID.....	775
8.6 获取集群 ID.....	776
8.7 自定义指标列表.....	777

1 使用前必读

概述

欢迎使用数据仓库服务GaussDB(DWS)。数据仓库服务是完全托管的企业级云上数据仓库服务，具备免运维、在线扩展、高效的多源数据加载能力，兼容PostgreSQL生态，并助力企业经济高效地对海量数据进行在线分析，实现数据快速变现。

您可以使用本文档提供的API对数据仓库服务的集群、快照、标签进行相关操作，如创建、查询、删除等，支持的全部操作请参见[API概述](#)。

在调用数据仓库服务API之前，请确保已经充分了解数据仓库服务相关概念，详细信息请参见[产品介绍](#)。

另外，数据仓库服务还提供了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](#)。

终端节点

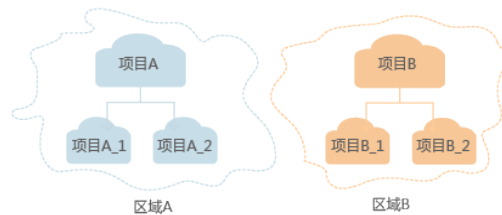
终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询所有服务的终端节点。

基本概念

- 账号
用户注册时的账号，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
指云资源所在的物理位置，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用户的要求，或满足不同地区的法律或其他要求。

- 可用区
一个可用区是一个或多个物理数据中心的集合，有独立的电力和网络，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，并在子项目中申请资源，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 概述

数据仓库服务所提供的接口为自研接口。通过配合使用数据仓库服务自研接口，您可以使用数据仓库服务的如下功能。

表 2-1 API 接口

类型	API	说明
集群管理接口	创建集群前检查	创建集群前预检查。
	创建集群	创建集群。
	查询集群列表	查询并显示集群列表。
	查询集群详情	查询集群详情。
	查询节点类型	查询GaussDB(DWS)服务支持的所有节点类型。
	删除集群	删除集群。
	重启集群	重启集群。
	扩容集群	扩容集群大小。
	重置密码	重置集群管理员密码。
	查询专属分布式存储池列表	查询租户可使用的专属分布式存储池。
	查询集群参数组	查询集群所关联的参数组。
	解除只读	当集群进入只读状态时，无法进行数据库相关操作，用户可以在管理控制台解除集群的只读状态。
	查询集群CN节点	查询指定集群的CN节点信息。
	批量增加CN节点	为指定集群批量增加CN节点。
	批量删除CN节点	为指定集群删除CN节点。
集群缩容	用于缩容集群。	

类型	API	说明
快照管理 接口	创建快照	为指定集群创建快照。
	查询快照列表	查询快照列表。
	查询快照详情	使用快照id查询快照详情。
	删除手动快照	删除一个指定快照。
	删除快照策略	删除一个快照策略。
	恢复快照到新集群	使用快照恢复集群。
	复制快照	复制一个自动快照。
	添加快照策略	设置快照策略。
事件管理	查询事件列表	查询事件列表。
	查询事件配置	查询事件配置。
	查询订阅事件	查询订阅的事件。
	创建订阅事件	添加订阅的事件。
	更新订阅事件	更新订阅事件。
	删除订阅事件	删除订阅的事件。
数据源	查询数据源	用于查询数据源。
	更新数据源	用于更新一个数据源。
审计日志	查询审计日志记录	查询审计日志记录。
资源管理	查询资源管理状态	查询资源管理状态。
	添加资源管理计划	添加资源管理计划。
	查询资源池列表	查询资源池。
	删除资源池	删除资源池。
	设置资源管理开关	设置资源管理。
告警管理	查询告警配置	查询告警配置。
	查询告警详情列表	查询告警详情列表。
	查询告警统计列表	查询告警统计。
	查询告警订阅列表	查询订阅告警。
	创建告警订阅	创建告警订阅。
	更新告警订阅	更新订阅的告警。
	删除告警订阅	删除订阅的告警。
可用区	查询可用区列表	查询租户创建集群时可使用的可用区列表。

类型	API	说明
连接管理	申请域名	为指定集群申请域名
	修改集群域名	为指定集群修改域名。
	删除集群域名	删除指定集群域名。
	获取集群可绑定的ELB列表	查询集群可以关联的ELB列表。
	集群绑定ELB	集群绑定ELB。
	集群解绑ELB	集群解绑ELB。
	集群绑定EIP	集群绑定ELP。
	集群解绑EIP	集群解绑EIP。
标签管理	查询项目标签	查询项目标签列表。
	查询集群标签	查询指定集群的标签信息。
	批量添加标签	为指定集群批量添加标签。
	批量删除标签	为指定集群批量删除标签。
配额管理	查询配额	查询单租户在GaussDB(DWS)服务下的配额信息。
容灾管理	查询容灾列表	查询容灾列表。

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[获取用户Token](#)来说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

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。

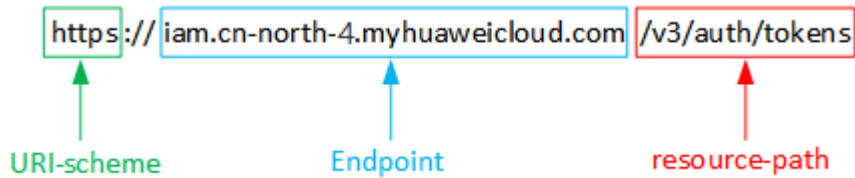
表 3-1 URI 中的参数说明

参数	描述
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GaussDB(DWS)在“华北-北京四”区域的Endpoint为“dws.cn-north-4.myhuaweicloud.com”。
resource-path	资源路径，即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
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。

例如您需要获取IAM在“华北-北京四”区域的Token，则需使用“华北-北京四”区域的Endpoint（iam.cn-north-4.myhuaweicloud.com），并在[获取用户Token](#)的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

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），它告诉服务您正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
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[获取用户Token](#)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/auth/tokens
```

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。

详细的公共请求消息头字段请参见[表3-3](#)。

表 3-3 公共请求消息头

名称	描述	是否必选	示例
x-sdk-date	请求的发生时间，格式为 (YYYYMMDD'T'HHMMSS'Z')。取值为当前系统的 GMT 时间。	否	20150907T101459Z
Host	请求的服务器信息，从服务 API 的 URL 中获取，值为 hostname[:port]。端口缺省时使用默认的端口，HTTPS 的默认端口为 443。	否	code.test.com or code.test.com:443
Content-Type	发送的实体的 MIME 类型。推荐用户默认使用 application/json，如果 API 是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。	是	application/json
Content-Length	请求 body 长度，单位为 Byte。	否	3495
X-Project-id	project id，项目 ID。请参考 获取项目 ID 章节获取项目 ID。	否	e9993fc787d94b6c886cb aa340f9c0f4
X-Auth-Token	用户 Token。 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。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为 Token 值。	否 使用 Token 认证时该字段必选。	以下仅为 Token 示例片段： MIIPAgYJKoZIhvcNAQcCo ...ggg1BBIINPXsidG9rZ
X-Language	请求语言。 支持配置如下值： <ul style="list-style-type: none"> zh-cn: 中文 en-us: 英文 	否	zh_cn

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-type”即可，添加消息头后的请求如下所示。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
Content-type: application/json
```

请求消息体

请求消息体通常以结构化格式（如JSON或XML）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。若请求消息体中的参数支持中文，则中文字符必须为UTF-8编码。

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**获取用户Token**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，加粗的斜体字段需要根据实际值填写，其中 ***user_name***为用户名，***domainname***为用户所属的账号名称，***********为用户登录密码，***xxxxxxxxxxxxxxxxxxxx***为project的名称，例如cn-north-4，可以从**地区和终端节点**中获取。请联系数据库管理员获取。

说明

scope参数定义了Token的作用域，上面示例中获取的Token仅能访问project下的资源。您还可以设置Token的作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见**获取用户Token**。

```
POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens
```

```
Content-type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "user_name",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "id": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用**curl**、**Postman**或直接编写代码等方式发送请求调用API。对于获取用户Token接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

3.2 认证鉴权

调用接口支持使用Token认证。

Token 认证

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。

📖 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。

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": "user_name",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "id": "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: ABCDEFJ....”加到请求消息头即可，如下所示。

```
GET https://iam.cn-north-4.myhuaweicloud.com/v3/auth/projects
Content-type: application/json
X-Auth-Token: ABCDEFJ....
```

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

状态码

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[获取用户Token](#)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[获取用户Token](#)接口，返回如[图3-2](#)所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 3-2 获取用户 Token 响应消息头

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → [REDACTED]
x-xss-protection → 1; mode=block;
```

响应消息体

响应消息体通常以结构化格式（如JSON或XML）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[获取用户Token](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "cn-north-4",
            .....

```

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": "The format of message is error",

```

```
"error_code": "AS.0001"  
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4 快速入门

本节通过调用一系列API，以集群管理为例，介绍使用GaussDB(DWS) API的基本流程。集群管理的流程如下：

1. 调用[认证鉴权](#)获取用户的Token，因为在后续的请求中需要将Token放到请求消息头中作为认证。
2. 调用[查询节点类型](#)接口获取支持的节点类型。
3. 调用[创建集群](#)接口创建集群。
4. 调用[查询集群列表](#)接口获取集群信息。
5. 调用[查询集群详情](#)接口查看集群详细内容。
6. 调用[创建快照](#)接口创建快照。
7. 调用[查询快照列表](#)接口查看快照是否创建成功。
8. 调用[恢复快照到新集群](#)接口使用快照恢复集群。
9. 当不再需要快照时，调用[删除手动快照](#)接口删除快照。
10. 当集群使用完成或不再需要时，调用[删除集群](#)接口删除集群。

前提条件

- 已创建VPC、子网和安全组，并获取VPC、子网和安全组的ID，具体请参见[创建VPC](#)。
- 已获取IAM的Endpoint和GaussDB(DWS)的Endpoint。
- 已获取项目ID，具体请参见[获取项目ID](#)。

集群管理

假设已获取前提条件中的信息为如下值（实际操作中您需要根据实际情况替换）：

- IAM的Endpoint：下面示例中以iam_endpoint代替
- GaussDB(DWS)的Endpoint：下面示例中以dws_endpoint代替
- VPC的ID：219ab8a0-1272-4049-a383-8ad0b770fa11
- 子网的ID：d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec
- 安全组的ID：12e3c23a-8710-4b75-95e4-5c8d7f68ef3c
- 项目ID：9bc552e6-19af-4326-800d-281a92984636

则可以通过如下步骤管理集群。

步骤1 调用其他接口前，需要调用[认证鉴权](#)中的接口获取Token，并设置成环境变量。

```
curl -H "Content-type:application/json" https://{iam_endpoint}/v3/auth/tokens -X POST -d '{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "testname",
          "domain": {
            "name": "testname"
          },
          "password": "Passw0rd"
        }
      }
    },
    "scope": {
      "project": {
        "name": "cn-north-4"
      }
    }
  }
}' -v -k
```

1. 获取响应Header中“X-Subject-Token”的值即为Token，示例如下：
X-Subject-Token:MlidkgYJKoZlhvcNAQcColidgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
2. 使用如下命令将token设置为环境变量。

```
export Token={X-Subject-Token}
```

X-Subject-Token即为上一步骤获取到的token，示例如下：

```
export Token=MlidkgYJKoZlhvcNAQcColidgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
```

步骤2 调用[查询节点类型](#)接口获取支持的节点类型。

```
curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/node_types -v -k
```

获取响应如下所示：

```
status CODE 200
{
  "node_types": [
    {
      "spec_name": "dws.d2.xlarge",
      "id": "ebe532d6-665f-40e6-a4d4-3c51545b6a67",
      "detail": [
        {
          "type": "vCPU",
          "value": "4"
        },
        {
          "value": "1675",
          "type": "LOCAL_DISK",
          "unit": "GB"
        },
        {
          "type": "mem",
          "value": "32",
          "unit": "GB"
        }
      ]
    },
    {
      "spec_name": "dws.m1.xlarge.ultrahigh",
      "id": "ebe532d6-665f-40e6-a4d4-3c51545b4f71",
      "detail": [
        {
          "type": "vCPU",
```

```

        "value": "4"
      },
      {
        "value": "512",
        "type": "SSD",
        "unit": "GB"
      },
      {
        "type": "mem",
        "value": "32",
        "unit": "GB"
      }
    ]
  }
}
]
}

```

步骤3 调用**创建集群**接口创建集群。

创建集群配置样例信息如下：

- 集群名称：dws-demo
- 管理员用户名：dbadmin
- 管理员密码：*passw0rd*
- 端口：8000
- 节点类型：dws.d1.xlarge
- 节点数量：3
- 弹性IP：auto_assign（自动绑定）

```

curl -X POST -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
  "node_type": "dws.d1.xlarge",
  "number_of_node": 3,
  "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
  "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
  "vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
  "port": 8000,
  "name": "dws-demo",
  "user_name": "dbadmin",
  "user_pwd": "passw0rd",
  "public_ip": {
    "public_bind_type": "auto_assign"
  }
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters -v -k

```

如果返回状态码为200，则说明创建集群请求下发成功。

步骤4 调用**查询集群列表**接口获取集群信息。

```

curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters -k -v

```

获取响应如下所示：

```

{
  "clusters": [
    {
      "id": "7ba031f6-81f4-4670-ad20-c490b91877e5",
      "status": "AVAILABLE",
      "sub_status": "NORMAL",
      "task_status": null,
      "action_progress": null,
      "node_type": "dws.d1.xlarge",
      "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
      "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
      "number_of_node": 3,
      "availability_zone": "cn-north-4b",

```

```

"port": 8000,
"name": "dws-demo",
"version": "1.1.0",
"vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
"user_name": "dbadmin",
"public_ip": {
  "public_bind_type": "auto_assign",
  "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
},
"public_endpoints": [
  {
    "public_connect_info": "dws-demo.cn-north-4.myhuaweicloud.com",
    "jdbc_url": "jdbc:postgresql://dws-demo.cn-north-4.myhuaweicloud.com/
<YOUR_DATABASE_name>"
  }
],
"endpoints": [
  {
    "connect_info": "dws-demo.cn-north-4.myhuaweicloud.com",
    "jdbc_url": "jdbc:postgresql://dws-demo.cn-north-4.myhuaweicloud.com/
<YOUR_DATABASE_name>"
  }
],
"updated": "2018-01-15T12:50:06",
"created": "2018-01-15T12:50:06",
"recent_event": 1
}
]
}

```

- status的状态如果为“CREATING”，则表示集群正处于创建中；若为“AVAILABLE”则表示集群创建成功。
- dws-demo集群对应的UUid为7ba031f6-81f4-4670-ad20-c490b91877e5，记录该UUid，方便后续步骤使用。

步骤5 调用[查询集群详情](#)接口查看集群详细内容。

```

curl -X GET -H "Content-type:application/json" -H "X-Auth-Token:$Token"
https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters/7ba031f6-81f4-4670-
ad20-c490b91877e5 -k -v

```

获取响应如下所示：

```

{
  "cluster": {
    "id": "7ba031f6-81f4-4670-ad20-c490b91877e5",
    "status": "AVAILABLE",
    "name": "dws-demo",
    "updated": "2018-01-15T12:50:06",
    "created": "2018-01-15T12:50:06",
    "user_name": "dbadmin",
    "sub_status": "NORMAL",
    "task_status": null,
    "action_progress": null,
    "node_type": "dws.d1.xlarge",
    "node_type_id": "5ddb1071-c5d7-40e0-a874-8a032e81a697",
    "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
    "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
    "number_of_node": 3,
    "availability_zone": "cn-north-4b",
    "port": 8000,
    "vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
    "public_ip": {
      "public_bind_type": "auto_assign",
      "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
    },
    "public_endpoints": [
      {
        "public_connect_info": "dws-demo.cn-north-4.myhuaweicloud.com",

```

```
      "jdbc_url": "jdbc:postgresql://dws-demo.cn-north-4.myhuaweicloud.com/  
<YOUR_DATABASE_name>"  
    },  
    ],  
    "endpoints": [  
      {  
        "connect_info": "dws-demo.cn-north-4.myhuaweicloud.com",  
        "jdbc_url": "jdbc:postgresql://dws-demo.cn-north-4.myhuaweicloud.com/  
<YOUR_DATABASE_name>"  
      },  
      {  
        "connect_info": "192.168.0.12:8000",  
        "jdbc_url": "jdbc:postgresql://192.168.0.12:8000/<YOUR_DATABASE_name>"  
      }  
    ],  
    "version": "1.1.0",  
    "maintain_window": {  
      "day": "Wed",  
      "start_time": "18:00",  
      "end_time": "22:00"  
    },  
    "recent_event": 1,  
    "tags": null,  
    "parameter_group": {  
      "id": "157e9cc4-64a8-11e8-adc0-fa7ae01bbebc",      "name": "Default-Parameter-Group-  
dws",  
      "status": "In-Sync"  
    }  
  }  
}
```

响应消息中还可以查询到集群的“public_endpoints”和“endpoints”等信息，集群创建成功后，可以使用“public_endpoints”或“endpoints”从外部进行集群访问。

步骤6 调用[创建快照](#)接口创建快照。

基于集群dws-demo创建一个名称为snapshotForDemoCluster的快照。

```
curl -X POST -H "Content-type:application/json" -H "X-Auth-Token:$Token" -d '{  
  "snapshot": {  
    "name": "snapshotForDemoCluster",  
    "cluster_id": "7ba031f6-81f4-4670-ad20-c490b91877e5",  
    "description": "Snapshot description"  
  }  
> https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots -k -v
```

获取响应如下所示：

```
{  
  "snapshot": {  
    "id": "2a4d0f86-67cd-408a-8b66-017454fb7793"  
  }  
}
```

如果返回状态码为200，则说明快照创建请求下发成功，记录“id”，供后续查询快照详情时使用。

步骤7 调用[查询快照列表](#)接口查看快照是否创建成功。

```
curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://  
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/  
2a4d0f86-67cd-408a-8b66-017454fb7793 -k -v
```

如果响应消息中快照状态为“AVAILABLE”，则说明快照创建成功；若为“CREATING”，则表示快照正在创建中。

```
{  
  "snapshot": {  
    "id": "2a4d0f86-67cd-408a-8b66-017454fb7793",  
    "name": "snapshotForDemoCluster",  
    "description": "Snapshot description",  
  }  
}
```



```

    "started": "2018-01-18T13:59:23Z",
    "finished": "2018-01-18T13:01:40Z",
    "size": 500,
    "status": "AVAILABLE",
    "type": "MANUAL",
    "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
  }
}

```

步骤8 调用[恢复快照到新集群](#)接口使用快照恢复集群。

将快照snapshotForDemoCluster恢复到名称为dws-restore的新集群。

```

curl -X POST -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
  "restore": {
    "name": "dws-restore"
  }
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/
2a4d0f86-67cd-408a-8b66-017454fb7793/actions -v -k

```

如果返回状态码为200，则说明集群恢复成功，可通过[恢复快照到新集群](#)查看集群恢复状态。

步骤9 当快照不再需要时，调用[删除手动快照](#)接口删除快照。

```

curl -X DELETE -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/
2a4d0f86-67cd-408a-8b66-017454fb7793 -v -k

```

如果返回状态码为202，则说明快照删除成功。

步骤10 当集群使用完成或不再需要时，调用[删除集群](#)接口删除集群。

```

curl -X DELETE -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
  "keep_last_manual_snapshot":0
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters/7ba031f6-81f4-4670-
ad20-c490b91877e5 -v -k

```

如果返回状态码为202，则说明集群删除成功。

----结束

5 API 说明

5.1 可用区

5.1.1 查询可用区列表

功能介绍

该接口用于查询租户创建集群时可使用的可用区列表。在创建实例时，需要配置实例所在的可用区ID。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/availability-zones

表 5-1 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

无

响应参数

状态码： 200

表 5-2 响应 Body 参数

参数	参数类型	描述
availability_zones	Array of AvailabilityZone objects	可用区列表对象。
count	Integer	可用区数量。

表 5-3 AvailabilityZone

参数	参数类型	描述
code	String	可用区唯一编码。
name	String	可用区名称。
status	String	可用区状态。 <ul style="list-style-type: none"> available: 正常。 unavailable: 不可用。
public_border_group	String	可用区组, 如: center。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/availability-zones
```

响应示例

状态码: 200

查询可用区列表成功。

```
{
  "availability_zones": [ {
    "code": "az1",
    "name": "AZ1",
    "status": "available",
    "public_border_group": "center"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAvailabilityZonesSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAvailabilityZonesRequest request = new ListAvailabilityZonesRequest();
        try {
            ListAvailabilityZonesResponse response = client.listAvailabilityZones(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

```
try:
    request = ListAvailabilityZonesRequest()
    response = client.list_availability_zones(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAvailabilityZonesRequest{}
    response, err := client.ListAvailabilityZones(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	查询可用区列表成功。
400	请求错误。

状态码	描述
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2 集群管理

5.2.1 创建集群

功能介绍

该接口用于创建集群。

集群必须要运行在VPC之内，创建集群前，您需要先创建VPC，并获取VPC和子网的ID。

该接口为异步接口，创建集群需要10~15分钟。暂不支持包年/包月计费模式。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters

表 5-4 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 5-5 请求 Body 参数

参数	是否必选	参数类型	描述
cluster	是	CreateClusterInfo object	集群对象。

表 5-6 CreateClusterInfo

参数	是否必选	参数类型	描述
node_type	是	String	节点类型。节点类型详情请参见 数据仓库规格 。 说明 跟据数据仓库规格详情，如果输入为存算一体规格，则创建的集群类型为存算一体集群；输入存算分离规格，则集群类型为存算分离集群。
number_of_node	是	Integer	集群节点数量，集群模式取值范围为3~256，实时数仓（单机模式）取值为1。
subnet_id	是	String	指定子网ID，用于集群网络配置。
security_group_id	是	String	指定安全组ID，用于集群网络配置。
vpc_id	是	String	指定虚拟私有云ID，用于集群网络配置。
availability_zone	否	String	配置集群可用区。集群可用区选择详情请参见 地区和终端节点 。 说明 暂不支持同时选择多个可用区。
port	否	Integer	集群服务端口，取值范围为8000~30000，默认值：8000。
name	是	String	集群名称，要求唯一性，必须以字母开头并只包含字母、数字、中划线或下划线，长度为4~64个字符且不超过64个字节，1个中文字符为3个字节，其他字符为1个字节。
user_name	是	String	DWS集群管理员用户名。用户命名要求如下： <ul style="list-style-type: none"> 只能由小写字母、数字或下划线组成。 必须由小写字母或下划线开头。 长度为6~64个字符。 用户名不能为DWS数据库的关键字。

参数	是否必选	参数类型	描述
user_pwd	是	String	DWS集群管理员密码。 <ul style="list-style-type: none"> 12~32个字符。 至少包含以下字符中的3种：大写字母、小写字母、数字和特殊字符（~!?,;:_(){}[]/<>@#%^&*+ \=-）。 不能与用户名或倒序的用户名相同。 说明 认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
public_ip	否	PublicIp object	公网IP地址。如果未指定，则默认不使用公网连接。
number_of_cn	否	Integer	CN部署量。取值范围为3~集群节点数，最大值为20，默认值为3，存算一体（单机模式）取值为1。
tags	否	Tags object	标签。
enterprise_project_id	否	String	企业项目ID。对集群指定企业项目，如果未指定，则使用默认企业项目“default”的ID，即0。

表 5-7 PublicIp

参数	是否必选	参数类型	描述
public_bind_type	否	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> auto_assign：现在购买。 not_use：暂不使用。 bind_existing：使用已有。
eip_id	否	String	弹性IP的ID。

表 5-8 Tags

参数	是否必选	参数类型	描述
key	是	String	键。输入标签键的最大长度为128个unicode字符，不能为空字符串，且首尾字符不能为空格。不能包含“=”、“*”、“<”、“>”、“\”、“,”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。
value	是	String	值。输入标签值的最大长度为255个字符，首尾字符不能为空格，可以为空字符串。不能包含“=”、“*”、“<”、“>”、“\”、“,”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。

响应参数

状态码： 200

表 5-9 响应 Body 参数

参数	参数类型	描述
cluster	Cluster object	集群对象。

表 5-10 Cluster

参数	参数类型	描述
id	String	集群ID。

请求示例

创建DWS集群，可用区为az1.dc1，节点规格为dws.m1.xlarge.ultrahigh，3节点，数据库端口8000，并设置公网IP，安全组ID，子网ID，虚拟私有云ID：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters
```

```
{
  "cluster": {
    "node_type": "dws.m1.xlarge.ultrahigh",
    "number_of_node": 3,
    "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
    "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    "availability_zone": "az1.dc1",
    "port": 8000,
  }
}
```

```
"name" : "dws-1",
"user_name" : "dbadmin",
"user_pwd" : "Passw0rd!",
"public_ip" : {
  "public_bind_type" : "auto_assign",
  "eip_id" : ""
},
"enterprise_project_id" : "aca4e50a-266f-4786-827c-f8d6cc3fbada"
}
```

响应示例

状态码： 200

创建集群成功。

```
{
  "cluster" : {
    "id" : "7d85f602-a948-4a30-afd4-e84f47471c15"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterRequestBody body = new CreateClusterRequestBody();
        PublicIp publicIpCluster = new PublicIp();
        publicIpCluster.withPublicBindType("auto_assign")
            .withEipId("");
        CreateClusterInfo clusterbody = new CreateClusterInfo();
        clusterbody.withNodeType("dws.m1.xlarge.ultrahigh")
    }
}
```

```
.withNumberOfNode(3)
.withSubnetId("374eca02-cfc4-4de7-8ab5-dbebf7d9a720")
.withSecurityGroupId("dc3ec145-9029-4b39-b5a3-ace5a01f772b")
.withVpcId("85b20d7e-9eb7-4b2a-98f3-3c8843ea3574")
.withAvailabilityZone("az1.dc1")
.withPort(8000)
.withName("dws-1")
.withUserName("dbadmin")
.withUserPwd("PasswOrd!")
.withPublicIp(publicIpCluster)
.withEnterpriseProjectId("aca4e50a-266f-4786-827c-f8d6cc3fbada");
body.withCluster(clusterbody);
request.withBody(body);
try {
    CreateClusterResponse response = client.createCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateClusterRequest()
        publicIpCluster = PublicIp(
            public_bind_type="auto_assign",
            eip_id=""
        )
        clusterbody = CreateClusterInfo(
            node_type="dws.m1.xlarge.ultrahigh",
            number_of_node=3,
            subnet_id="374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
            security_group_id="dc3ec145-9029-4b39-b5a3-ace5a01f772b",
            vpc_id="85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
            availability_zone="az1.dc1",
            port=8000,
```

```
        name="dws-1",
        user_name="dbadmin",
        user_pwd="Passw0rd!",
        public_ip=publicIpCluster,
        enterprise_project_id="aca4e50a-266f-4786-827c-f8d6cc3fbada"
    )
    request.body = CreateClusterRequestBody(
        cluster=clusterbody
    )
    response = client.create_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    eipIdPublicIp := ""
    publicIpCluster := &model.PublicIp{
        PublicBindType: "auto_assign",
        EipId: &eipIdPublicIp,
    }
    availabilityZoneCluster := "az1.dc1"
    portCluster := int32(8000)
    enterpriseProjectIdCluster := "aca4e50a-266f-4786-827c-f8d6cc3fbada"
    clusterbody := &model.CreateClusterInfo{
        NodeType: "dws.m1.xlarge.ultrahigh",
        NumberOfNode: int32(3),
        SubnetId: "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
        SecurityGroupId: "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
        VpcId: "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
        AvailabilityZone: &availabilityZoneCluster,
        Port: &portCluster,
        Name: "dws-1",
        UserName: "dbadmin",
        UserPwd: "Passw0rd!",
    }
```

```

PublicIp: publicIpCluster,
EnterpriseProjectId: &enterpriseProjectIdCluster,
}
request.Body = &model.CreateClusterRequestBody{
Cluster: clusterbody,
}
response, err := client.CreateCluster(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	创建集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.2.2 V2 创建集群

功能介绍

该接口用于创建集群。集群必须要运行在VPC之内，创建集群前，您需要先创建VPC，并获取VPC和子网的ID。

说明

该接口为异步接口，创建集群需要10~15分钟。暂不支持包年/包月计费模式。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters

表 5-11 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 5-12 请求 Body 参数

参数	是否必选	参数类型	描述
cluster	否	V2CreateCluster object	创建集群请求。

表 5-13 V2CreateCluster

参数	是否必选	参数类型	描述
name	是	String	集群名称，要求唯一性，必须以字母开头并只包含字母、数字、中划线或下划线，长度为4~64个字符且不超过64个字节，1个中文字符为3个字节，其他字符为1个字节。
flavor	是	String	集群规格名称。节点规格详情请参见 数据仓库规格 。
num_cn	是	Integer	集群CN数量，取值范围为3~集群节点数，最大值为20，默认值为3，存算一体（单机模式）取值为1。
num_node	是	Integer	集群节点数量，集群模式取值范围为3~256，存算一体（单机模式）取值为1。
db_name	是	String	管理员用户名称。用户命名要求如下： <ul style="list-style-type: none"> 只能由小写字母、数字或下划线组成。 必须由小写字母或下划线开头。 长度为6~64个字符。 用户名不能为DWS数据库的关键字。

参数	是否必选	参数类型	描述
db_password	是	String	<p>管理员用户密码。</p> <ul style="list-style-type: none"> 12~32个字符 至少包含以下字符中的3种：大写字母、小写字母、数字和特殊字符（~!?,;:_(){}[]/<>@#%^&*+ \=-）。 不能与用户名或倒序的用户名相同。 <p>说明 认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。</p>
db_port	是	Integer	集群数据库端口，取值范围为8000~30000，默认值：8000。
dss_pool_id	否	String	专属存储池ID。
availability_zones	是	Array of String	<p>可用区列表。集群可用区选择详情请参见地区和终端节点。</p> <p>说明 当前多AZ集群只支持选择3个AZ。</p>
tags	否	Array of Tags objects	标签。
vpc_id	是	String	指定虚拟私有云ID，用于集群网络配置。
subnet_id	是	String	指定子网ID，用于集群网络配置。
security_group_id	否	String	指定安全组ID，用于集群网络配置。
public_ip	否	PublicIp object	公网IP地址，如果未指定，则默认不使用公网连接。
datastore_version	是	String	集群版本。
master_key_id	否	String	密钥ID。
master_key_name	否	String	密钥名称。
crypt_algorithm	否	String	加密算法。
volume	是	Volume object	磁盘信息。根据集群规格名称判断是云盘规格还是本地盘规格，云盘规格必选，需要配置磁盘名称与磁盘容量；本地盘规格非必填。

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目ID，对集群指定企业项目，如果未指定，则使用默认企业项目“default”的ID，即0。
ipv6_enable	否	boolean	指定网络协议类型，表明是否支持IPv6，默认不使用IPv6。

表 5-14 Tags

参数	是否必选	参数类型	描述
key	是	String	键。输入标签键的最大长度为128个unicode字符，不能为空字符串，且首尾字符不能为空格。不能包含“=”，“*”，“<”，“>”，“\”，“，”，“ ”，“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。
value	是	String	值。输入标签值的最大长度为255个字符，首尾字符不能为空格，可以为空字符串。不能包含“=”，“*”，“<”，“>”，“\”，“，”，“ ”，“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。

表 5-15 PublicIp

参数	是否必选	参数类型	描述
public_bind_type	否	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> • auto_assign：自动绑定 • not_use：暂未使用 • bind_existing：使用已有
eip_id	否	String	弹性IP的ID。

表 5-16 Volume

参数	是否必选	参数类型	描述
volume	是	String	磁盘名称，取值如下： <ul style="list-style-type: none"> SSD（超高IO） 高IO（SAS） 普通IO（SATA）
capacity	否	Integer	磁盘容量，单位：GB。

响应参数

状态码： 200

表 5-17 响应 Body 参数

参数	参数类型	描述
cluster	Cluster object	集群对象。

表 5-18 Cluster

参数	参数类型	描述
id	String	集群ID。

请求示例

创建多AZ DWS集群，集群版本为8.2.1，可用区是az1、az2和az3，节点规格为dws.m1.xlarge.ultrahigh，3节点，SSD磁盘，容量为100GB，数据库端口为8000，带公网IP，安全组ID，子网ID，虚拟私有云ID：

POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters

```
{
  "cluster": {
    "flavor": "dws.m1.xlarge.ultrahigh",
    "num_node": 3,
    "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
    "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    "db_port": 8000,
    "datastore_version": "8.2.1",
    "availability_zones": [ "az1", "az2", "az3" ],
    "name": "dws-1",
    "num_cn": 3,
    "db_name": "dbadmin",
    "db_password": "Passw0rd!",
    "public_ip": {
      "public_bind_type": "auto_assign",
      "eip_id": ""
    }
  }
}
```

```
"volume":{
  "volume":"SSD",
  "capacity":100
},
"enterprise_project_id" : "aca4e50a-266f-4786-827c-f8d6cc3fbada"
}
}
```

响应示例

状态码： 200

创建集群成功。

```
{
  "cluster" : {
    "id" : "7d85f602-a948-4a30-afd4-e84f47471c15"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class CreateClusterV2Solution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateClusterV2Request request = new CreateClusterV2Request();
        V2CreateClusterReq body = new V2CreateClusterReq();
        PublicIps publicIpsCluster = new PublicIps();
        publicIpsCluster.withPublicBindType("auto_assign")
            .withEipId("");
        List<String> listClusterAvailabilityZones = new ArrayList<>();
        listClusterAvailabilityZones.add("az1");
        listClusterAvailabilityZones.add("az2");
    }
}
```

```
V2CreateCluster clusterbody = new V2CreateCluster();
clusterbody.setName("dws-1")
    .withFlavor("dws.m1.xlarge.ultrahigh")
    .withNumNode(3)
    .withDbName("dbadmin")
    .withDbPassword("Passw0rd!")
    .withDbPort(8000)
    .withAvailabilityZones(listClusterAvailabilityZones)
    .withVpclid("85b20d7e-9eb7-4b2a-98f3-3c8843ea3574")
    .withSubnetId("374eca02-cfc4-4de7-8ab5-dbebf7d9a720")
    .withSecurityGroupId("dc3ec145-9029-4b39-b5a3-ace5a01f772b")
    .withPublicIp(publicIpCluster)
    .withDatastoreVersion("8.2.1")
    .withEnterpriseProjectId("aca4e50a-266f-4786-827c-f8d6cc3fbada");
body.withCluster(clusterbody);
request.withBody(body);
try {
    CreateClusterV2Response response = client.createClusterV2(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateClusterV2Request()
        publicIpCluster = PublicIp(
            public_bind_type="auto_assign",
            eip_id=""
        )
        listAvailabilityZonesCluster = [
            "az1",
            "az2"
        ]
        clusterbody = V2CreateCluster(
```

```
name="dws-1",
flavor="dws.m1.xlarge.ultrahigh",
num_node=3,
db_name="dbadmin",
db_password="Passw0rd!",
db_port=8000,
availability_zones=listAvailabilityZonesCluster,
vpc_id="85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
subnet_id="374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
security_group_id="dc3ec145-9029-4b39-b5a3-ace5a01f772b",
public_ip=publicIpCluster,
datastore_version="8.2.1",
enterprise_project_id="aca4e50a-266f-4786-827c-f8d6cc3fbada"
)
request.body = V2CreateClusterReq(
    cluster=clusterbody
)
response = client.create_cluster_v2(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterV2Request{}
    eipIdPublicIp := ""
    publicIpCluster := &model.PublicIp{
        PublicBindType: "auto_assign",
        EipId: &eipIdPublicIp,
    }
    var listAvailabilityZonesCluster = []string{
        "az1",
        "az2",
    }
    securityGroupIdCluster := "dc3ec145-9029-4b39-b5a3-ace5a01f772b"
    enterpriseProjectIdCluster := "aca4e50a-266f-4786-827c-f8d6cc3fbada"
```

```

clusterbody := &model.V2CreateCluster{
    Name: "dws-1",
    Flavor: "dws.m1.xlarge.ultrahigh",
    NumNode: int32(3),
    DbName: "dbadmin",
    DbPassword: "Passw0rd!",
    DbPort: int32(8000),
    AvailabilityZones: listAvailabilityZonesCluster,
    VpcId: "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    SubnetId: "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    SecurityGroupId: &securityGroupIdCluster,
    PublicIp: publicIpCluster,
    DatastoreVersion: "8.2.1",
    EnterpriseProjectId: &enterpriseProjectIdCluster,
}
request.Body = &model.V2CreateClusterReq{
    Cluster: clusterbody,
}
response, err := client.CreateClusterV2(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	创建集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.2.3 创建逻辑集群

功能介绍

该接口用于创建逻辑集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/logical-clusters

表 5-19 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-20 请求 Body 参数

参数	是否必选	参数类型	描述
logical_cluster	是	CreateLogicalClusterInfo object	创建逻辑集群信息。

表 5-21 CreateLogicalClusterInfo

参数	是否必选	参数类型	描述
logical_cluster_name	是	String	逻辑集群名称。
cluster_rings	是	Array of ClusterRing objects	逻辑集群环信息。

表 5-22 ClusterRing

参数	是否必选	参数类型	描述
ring_hosts	是	Array of RingHost objects	集群主机信息。

表 5-23 RingHost

参数	是否必选	参数类型	描述
host_name	是	String	主机名称。
back_ip	是	String	后端IP地址。
cpu_cores	是	Integer	主机CPU核数。
memory	是	Double	主机内存。
disk_size	是	Double	主机磁盘大小。

响应参数

状态码： 200

表 5-24 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

创建名为“v3_logical”的逻辑集群：

POST https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters

```
{
  "logical_cluster": {
    "logical_cluster_name": "v3_logical",
    "cluster_rings": [ {
      "ring_hosts": [ {
        "host_name": "host-172-16-20-246",
        "back_ip": "172.16.73.90",
        "cpu_cores": 8,
        "memory": 32.0,
        "disk_size": 800.0
      }, {
        "host_name": "host-172-16-4-26",
        "back_ip": "172.16.123.5",
        "cpu_cores": 8,
        "memory": 32.0,
        "disk_size": 800.0
      }, {
        "host_name": "host-172-16-4-26",
        "back_ip": "172.16.123.5",
        "cpu_cores": 8,
        "memory": 32.0,
        "disk_size": 800.0
      }
    ]
  }
}
```

响应示例

状态码： 200

创建逻辑集群成功。

```
{
  "error_code" : "DWS.0000",
  "error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class CreateLogicalClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateLogicalClusterRequest request = new CreateLogicalClusterRequest();
        CreateLogicalClusterRequestBody body = new CreateLogicalClusterRequestBody();
        List<RingHost> listClusterRingsRingHosts = new ArrayList<>();
        listClusterRingsRingHosts.add(
            new RingHost()
                .withHostName("host-172-16-20-246")
                .withBackIp("172.16.73.90")
                .withCpuCores(8)
                .withMemory((double)32.0)
                .withDiskSize((double)800.0)
        );
        listClusterRingsRingHosts.add(
            new RingHost()
                .withHostName("host-172-16-4-26")
                .withBackIp("172.16.123.5")
                .withCpuCores(8)
                .withMemory((double)32.0)
        );
    }
}
```



```
        .withDiskSize((double)800.0)
    );
    listClusterRingsRingHosts.add(
        new RingHost()
            .withHostName("host-172-16-4-26")
            .withBackIp("172.16.123.5")
            .withCpuCores(8)
            .withMemory((double)32.0)
            .withDiskSize((double)800.0)
    );
    List<ClusterRing> listLogicalClusterClusterRings = new ArrayList<>();
    listLogicalClusterClusterRings.add(
        new ClusterRing()
            .withRingHosts(listClusterRingsRingHosts)
    );
    CreateLogicalClusterInfo logicalClusterbody = new CreateLogicalClusterInfo();
    logicalClusterbody.withLogicalClusterName("v3_logical")
        .withClusterRings(listLogicalClusterClusterRings);
    body.withLogicalCluster(logicalClusterbody);
    request.withBody(body);
    try {
        CreateLogicalClusterResponse response = client.createLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateLogicalClusterRequest()
        listRingHostsClusterRings = [
            RingHost(
                host_name="host-172-16-20-246",
                back_ip="172.16.73.90",
                cpu_cores=8,
```

```

        memory=32.0,
        disk_size=800.0
    ),
    RingHost(
        host_name="host-172-16-4-26",
        back_ip="172.16.123.5",
        cpu_cores=8,
        memory=32.0,
        disk_size=800.0
    ),
    RingHost(
        host_name="host-172-16-4-26",
        back_ip="172.16.123.5",
        cpu_cores=8,
        memory=32.0,
        disk_size=800.0
    )
]
listClusterRingsLogicalCluster = [
    ClusterRing(
        ring_hosts=listRingHostsClusterRings
    )
]
logicalClusterbody = CreateLogicalClusterInfo(
    logical_cluster_name="v3_logical",
    cluster_rings=listClusterRingsLogicalCluster
)
request.body = CreateLogicalClusterRequestBody(
    logical_cluster=logicalClusterbody
)
response = client.create_logical_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build()
    )
}

```

```

request := &model.CreateLogicalClusterRequest{}
var listRingHostsClusterRings = []model.RingHost{
    {
        HostName: "host-172-16-20-246",
        BackIp: "172.16.73.90",
        CpuCores: int32(8),
        Memory: float64(32.0),
        DiskSize: float64(800.0),
    },
    {
        HostName: "host-172-16-4-26",
        BackIp: "172.16.123.5",
        CpuCores: int32(8),
        Memory: float64(32.0),
        DiskSize: float64(800.0),
    },
    {
        HostName: "host-172-16-4-26",
        BackIp: "172.16.123.5",
        CpuCores: int32(8),
        Memory: float64(32.0),
        DiskSize: float64(800.0),
    },
}
var listClusterRingsLogicalCluster = []model.ClusterRing{
    {
        RingHosts: listRingHostsClusterRings,
    },
}
logicalClusterbody := &model.CreateLogicalClusterInfo{
    LogicalClusterName: "v3_logical",
    ClusterRings: listClusterRingsLogicalCluster,
}
request.Body = &model.CreateLogicalClusterRequestBody{
    LogicalCluster: logicalClusterbody,
}
response, err := client.CreateLogicalCluster(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	创建逻辑集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.2.4 物理集群转换到逻辑集群

功能介绍

该接口用于将物理集群转换为逻辑集群。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/clusters/{cluster_id}/convert-to-logical-cluster/{name}
```

表 5-25 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
name	是	String	逻辑集群名称。

请求参数

无

响应参数

状态码： 200

表 5-26 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
POST https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/convert-to-logical-cluster/elastic_group
```

响应示例

状态码： 200

转换到逻辑集群成功。

```
{
  "error_code" : "DWS.0000",
  "error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ConvertToLogicalClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ConvertToLogicalClusterRequest request = new ConvertToLogicalClusterRequest();
        try {
            ConvertToLogicalClusterResponse response = client.convertToLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ConvertToLogicalClusterRequest()
        response = client.convert_to_logical_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ConvertToLogicalClusterRequest{}
    response, err := client.ConvertToLogicalCluster(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	转换到逻辑集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.2.5 查询集群列表

功能介绍

该接口用于查询并显示集群列表。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/clusters
```

表 5-27 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

无

响应参数

状态码： 200

表 5-28 响应 Body 参数

参数	参数类型	描述
clusters	Array of ClusterInfo objects	集群对象列表。
count	Integer	集群对象的总数。

表 5-29 ClusterInfo

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
status	String	<p>集群状态，有效值包括：</p> <ul style="list-style-type: none"> ● CREATING (100)：创建中 ● ACTIVE (200)：可用 ● FAILED (300)：不可用 ● CREATION FAILED (303)：创建失败 ● DELETING (304)：删除中 ● DELETION FAILED (302)：删除失败 ● DELETED (400)：已删除 ● STOPING (910)：停止中 ● STOPPED (900)：已停止 ● STARTING (920)：启动中 ● FROZEN (800)：已冻结 <p>说明 如果用户查询返回状态为“AVAILABLE (200)”，也表示可用。</p>
version	String	数据仓库版本。
updated	String	集群上次修改时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ
created	String	集群创建时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ
port	Integer	集群服务端口。

参数	参数类型	描述
endpoints	Array of Endpoints objects	集群的内网连接信息。
nodes	Array of Nodes objects	集群实例。
tags	Array of Tags objects	集群标签。
user_name	String	管理员用户名。
number_of_node	Integer	集群节点数量。
recent_event	Integer	事件数。
availability_zone	String	可用区。
enterprise_project_id	String	企业项目ID。值为0表示默认企业项目“default”。
node_type	String	节点类型。
vpc_id	String	虚拟私有云ID。
subnet_id	String	子网ID。
public_ip	PublicIp object	公网IP地址，如果未指定，则默认不显示公网连接。
public_endpoints	Array of PublicEndpoints objects	集群的公网连接信息，如果未指定，则默认不显示公网连接信息。

参数	参数类型	描述
action_progress	Map<String,String>	<p>任务信息，由key、value组成。key值为正在进行的任务，value值为正在进行任务的进度。key值的有效值包括：</p> <ul style="list-style-type: none"> • CREATING：创建中 • RESTORING：恢复中 • SNAPSHOTTING：快照中 • GROWING：扩容中 • REBOOTING：重启中 • SETTING_CONFIGURATION：安全设置配置中 • CONFIGURING_EXT_DATASOURCE：MRS连接配置中 • ADD_CN_ING：增加CN中 • DEL_CN_ING：删除CN中 • REDISTRIBUTING：重分布中 • ELB_BINDING：弹性负载均衡绑定中 • ELB_UNBINDING：弹性负载均衡解绑中 • ELB_SWITCHING：弹性负载均衡切换中 • NETWORK_CONFIGURING：网络配置中 • DISK_EXPANDING：磁盘扩容中 • ACTIVE_STANDY_SWITCHOVER：主备恢复中 • CLUSTER_SHRINKING：缩容中 • SHRINK_CHECKING：缩容检测中 • FLAVOR_RESIZING：规格变更中 • MANAGE_IP_BINDING：登录开通中 • FINE_GRAINED_RESTORING：细粒度恢复中 • DR_RECOVERING：容灾恢复中 • REPAIRING：修复中 <p>示例如下： <pre>"action_progress": {"SNAPSHOTTING":"16%"}</pre></p>

参数	参数类型	描述
sub_status	String	<p>“可用”集群状态的子状态，有效值包括：</p> <ul style="list-style-type: none"> ● NORMAL：正常 ● READONLY：只读 ● REDISTRIBUTING：重分布中 ● REDISTRIBUTION-FAILURE：重分布失败 ● UNBALANCED：非均衡 ● UNBALANCED READONLY：非均衡，只读- ● DEGRADED：节点故障 ● DEGRADED READONLY：节点故障，只读 ● DEGRADED UNBALANCED：节点故障，非均衡 ● UNBALANCED REDISTRIBUTING：非均衡，重分布中 ● UNBALANCED REDISTRIBUTION-FAILURE：非均衡，重分布失败 ● READONLY REDISTRIBUTION-FAILURE：只读，重分布失败 ● UNBALANCED READONLY REDISTRIBUTION-FAILURE：非均衡，只读，重分布失败 ● DEGRADED REDISTRIBUTION-FAILURE：节点故障，重分布失败 ● DEGRADED UNBALANCED REDISTRIBUTION-FAILURE：节点故障，非均衡，只读，重分布失败 ● DEGRADED UNBALANCED READONLY REDISTRIBUTION-FAILURE：节点故障，非均衡，只读，重分布失败 ● DEGRADED UNBALANCED READONLY：节点故障，非均衡，只读

参数	参数类型	描述
task_status	String	<p>集群管理任务，有效值包括：</p> <ul style="list-style-type: none"> ● UNFREEZING：解冻中 ● FREEZING：冻结中 ● RESTORING：恢复中 ● SNAPSHOTTING：快照中 ● GROWING：扩容中 ● REBOOTING：重启中 ● SETTING_CONFIGURATION：安全设置配置中 ● CONFIGURING_EXT_DATASOURCE：MRS连接配置中 ● DELETING_EXT_DATASOURCE：删除MRS连接 ● REBOOT_FAILURE：重启失败 ● RESIZE_FAILURE：扩容失败 ● ADD_CN_ING：增加CN中 ● DEL_CN_ING：删除CN中 ● CREATING_NODE：添加节点 ● CREATE_NODE_FAILED：添加节点失败 ● DELETING_NODE：删除节点 ● DELETE_NODE_FAILED：删除节点失败 ● REDISTRIBUTING：重分布中 ● REDISTRIBUTE_FAILURE：重分布失败 ● WAITING_REDISTRIBUTION：待重分布 ● REDISTRIBUTION_PAUSED：重分布暂停 ● ELB_BINDING：弹性负载均衡绑定中 ● ELB_BIND_FAILED：弹性负载均衡绑定失败 ● ELB_UNBINDING：弹性负载均衡解绑中 ● ELB_UNBIND_FAILED：弹性负载均衡解绑失败 ● ELB_SWITCHING：弹性负载均衡切换中 ● ELB_SWITCHING_FAILED：弹性负载均衡切换失败 ● NETWORK_CONFIGURING：网络配置中 ● NETWORK_CONFIG_FAILED：网络配置失败 ● DISK_EXPAND_FAILED：磁盘扩容失败 ● DISK_EXPANDING：磁盘扩容中 ● ACTIVE_STANDY_SWITCHOVER：主备恢复中

参数	参数类型	描述
		<ul style="list-style-type: none"> ACTIVE_STANDY_SWITCHOVER_FAILURE: 主备恢复失败 CLUSTER_SHRINK_FAILED: 缩容失败 CLUSTER_SHRINKING: 缩容中 SHRINK_CHECK_FAILED: 缩容检测失败 SHRINK_CHECKING: 缩容检测中 FLAVOR_RESIZING_FAILED: 规格变更失败 FLAVOR_RESIZING: 规格变更中 MANAGE_IP_BIND_FAILED: 登录开通失败 MANAGE_IP_BINDING: 登录开通中 ORDER_PENDING: 订单待支付 FINE_GRAINED_RESTORING: 细粒度恢复中 DR_RECOVERING: 容灾恢复中
security_group_id	String	安全组ID。
failed_reasons	FailedReason object	失败原因。如果为空，则集群处于正常状态。

表 5-30 Endpoints

参数	参数类型	描述
connect_info	String	内网连接信息。
jdbc_url	String	内网JDBC URL，默认格式如下： jdbc:postgresql://<connect_info>/<YOUR_DATABASE_NAME>

表 5-31 Nodes

参数	参数类型	描述
id	String	集群实例ID。

参数	参数类型	描述
status	String	集群实例状态。 <ul style="list-style-type: none"> • 100: 创建中 • 199: 空闲 • 200: 可用 • 300: 不可用 • 303: 创建失败 • 304: 删除中 • 305: 删除失败 • 400: 已删除

表 5-32 Tags

参数	参数类型	描述
key	String	键。输入标签键的最大长度为128个字符，不能为空字符串，且首尾字符不能为空格。不能包含“=”、“*”、“<”、“>”、“\”、“'”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。
value	String	值。输入标签值的最大长度为255个字符，首尾字符不能为空格，可以为空字符串。不能包含“=”、“*”、“<”、“>”、“\”、“'”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。

表 5-33 PublicIp

参数	参数类型	描述
public_bind_type	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> • auto_assign: 现在购买 • not_use: 暂不使用 • bind_existing: 使用已有
eip_id	String	弹性IP的ID。

表 5-34 PublicEndpoints

参数	参数类型	描述
public_connect_info	String	公网连接信息。
jdbc_url	String	公网JDBC URL，默认格式如下： jdbc:postgresql://<public_connect_info>/<YOUR_DATABASE_name>

表 5-35 FailedReason

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters
```

响应示例

状态码： 200

查询集群列表成功。

```
{
  "clusters": [ {
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
    "name": "dws-1",
    "status": "AVAILABLE",
    "version": "1.2.0",
    "updated": "2016-02-10T14:28:14Z",
    "created": "2016-02-10T14:26:14Z",
    "port": 8000,
    "endpoints": [ {
      "connect_info": "dws-1.cn-north-4.myhuaweicloud.com",
      "jdbc_url": "jdbc:postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
    } ],
    "nodes": [ {
      "id": "acaf62a4-41b3-4106-bf6b-2f669d88291e",
      "status": "200"
    }, {
      "id": "d32de51e-4fcd-4e5a-a9dc-bb903abb494b",
      "status": "200"
    }, {
      "id": "d71a4a25-c9bc-4ffd-9f4a-e422aef327f9",
      "status": "200"
    } ],
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value2"
    } ],
    "user_name": "dbadmin",
  } ]
}
```

```
"number_of_node" : 3,
"recent_event" : 6,
"availability_zone" : "cn-north-4b",
"enterprise_project_id" : "0",
"node_type" : "dws.d1.xlarge.ultrahigh",
"vpc_id" : "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
"subnet_id" : "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
"public_ip" : {
  "public_bind_type" : "auto_assign",
  "eip_id" : "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
},
"public_endpoints" : [ {
  "public_connect_info" : "dws-1.cn-north-4.myhuaweicloud.com",
  "jdbc_url" : "jdbc.postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
} ],
"action_progress" : {
  "SNAPSHOTTING" : "20%"
},
"sub_status" : "READONLY",
"task_status" : "SNAPSHOTTING",
"security_group_id" : "dc3ec145-9029-4b39-b5a3-ace5a01f772b"
} ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClustersSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListClustersRequest request = new ListClustersRequest();
        try {
            ListClustersResponse response = client.listClusters(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.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
}

```

Python

```

# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListClustersRequest()
        response = client.list_clusters(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.ListClustersRequest{}
response, err := client.ListClusters(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	查询集群列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.6 查询逻辑集群列表

功能介绍

该接口用于查询逻辑集群列表。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v2/{project_id}/clusters/{cluster_id}/logical-clusters
```

表 5-36 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-37 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页查询，偏移。
limit	否	Integer	分页查询，每页显示的条目数量，默认值为10。

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
logical_clusters	Array of LogicalClusterInfo objects	逻辑集群列表信息。
count	Integer	逻辑集群总数量。
add_enable	Boolean	是否可以添加逻辑集群。 <ul style="list-style-type: none"> • true: 表示可以添加逻辑集群。 • false: 如果集群内有其他运维操作，该值为 false，此时不能添加逻辑集群。

表 5-39 LogicalClusterInfo

参数	参数类型	描述
logical_cluster_id	String	逻辑集群ID。

参数	参数类型	描述
logical_cluster_name	String	逻辑集群名称。
cluster_rings	Array of ClusterRing objects	逻辑集群主机信息。
status	String	逻辑集群状态。状态如下所示： <ul style="list-style-type: none"> Failed: 不可用 Normal: 可用 Unavailable: 不可用 Redistributing: 重分布 Redistribute_failed: 重分布失败 Unbalanced: 非均衡 Stopped: 已停止
first_logical_cluster	Boolean	是否为第一个逻辑集群。第一个创建或者转换的逻辑集群不能删除，因为其中包含了一些系统视图。 该字段仅在低版本下有意义，8.2.0及以上的集群版本在有多个逻辑集群时，第一个逻辑集群也支持删除，均返回false。
action_info	ActionInfo object	逻辑集群操作信息。
edit_enable	Boolean	是否允许编辑。
restart_enable	Boolean	是否允许重启。
delete_enable	Boolean	是否允许删除。
add_to_elastic	Boolean	是否允许弹性伸缩。
mode	String	逻辑集群模式。
waiting_for_killing	Integer	等待被销毁。
cluster_type	String	集群类型。

表 5-40 ClusterRing

参数	参数类型	描述
ring_hosts	Array of RingHost objects	集群主机信息。

表 5-41 RingHost

参数	参数类型	描述
host_name	String	主机名称。
back_ip	String	后端IP地址。
cpu_cores	Integer	主机CPU核数。
memory	Double	主机内存。
disk_size	Double	主机磁盘大小。

表 5-42 ActionInfo

参数	参数类型	描述
action_name	String	操作名称。当前允许操作如下： <ul style="list-style-type: none"> • Create: 创建 • Expand: 扩容 • Restart: 重启 • Delete: 删除 • Shrink: 缩容
progress	Integer	操作进度，默认10。
completed	Boolean	是否完成操作。
start_time	String	操作开始时间。
end_time	String	操作结束时间。
result	String	操作结果。success或者failed，默认空字符串。
logs	String	操作日志。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters
```

响应示例

状态码： 200

查询集群列表成功。

```
{
  "logical_clusters": [ {
    "logical_cluster_id": "fa9d69a0-3814-4fb9-98df-d9f4f5ef53ca",
    "logical_cluster_name": "elastic_group",
    "cluster_rings": [ {
      "ring_hosts": [ {
        "host_name": "host-172-16-20-246",
```

```
"back_ip" : "172.16.73.90",
"cpu_cores" : 8,
"memory" : 32.0,
"disk_size" : 800.0
}, {
"host_name" : "host-172-16-4-26",
"back_ip" : "172.16.123.5",
"cpu_cores" : 8,
"memory" : 32.0,
"disk_size" : 800.0
}, {
"host_name" : "host-172-16-43-90",
"back_ip" : "172.16.92.175",
"cpu_cores" : 8,
"memory" : 32.0,
"disk_size" : 800.0
}
]
}],
"status" : "Normal",
"first_logical_cluster" : false,
"action_info" : {
"progress" : 100,
"completed" : true,
"result" : "success",
"logs" : null,
"action_name" : "Grow",
"start_time" : "2023-06-05 01:58:43",
"end_time" : "2023-06-05 02:11:50"
},
"edit_enable" : true,
"restart_enable" : true,
"delete_enable" : false,
"add_to_elastic" : true,
"mode" : null,
"cluster_type" : null
}],
"count" : 1,
"add_enable" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListLogicalClustersSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListLogicalClustersRequest request = new ListLogicalClustersRequest();
request.withOffset(<offset>);
request.withLimit(<limit>);
try {
    ListLogicalClustersResponse response = client.listLogicalClusters(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListLogicalClustersRequest()
        request.offset = <offset>
        request.limit = <limit>
        response = client.list_logical_clusters(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListLogicalClustersRequest{}
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListLogicalClusters(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	查询集群列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.2.7 查询逻辑集群任务信息

功能介绍

该接口用于查询逻辑集群任务信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/logical-clusters/tasks

表 5-43 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-44 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页查询，偏移。
limit	否	Integer	分页查询，每页显示的条目数量。
logical_cluster_name	否	String	集群名称。
type	否	String	类型。
order_by	否	String	排序字段。
order	否	String	排序：升序/降序。

请求参数

无

响应参数

状态码： 200

表 5-45 响应 Body 参数

参数	参数类型	描述
logical_cluster_tasks	Array of LogicalClusterTaskInfo objects	逻辑集群任务信息。
count	Long	逻辑集群任务总数。

表 5-46 LogicalClusterTaskInfo

参数	参数类型	描述
type	String	任务类型。
logical_cluster_name	String	逻辑集群名称。
start_time	String	任务开始时间。
end_time	String	任务结束时间。
result	String	任务执行结果， success/failed。
log	String	任务执行日志。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters/tasks?offset=0&limit=10&logical_cluster_name=test_logical&type=Expand&order_by=startTime&order=DESC
```

响应示例

状态码： 200

查询任务信息成功。

```
{
  "logical_cluster_tasks": [ {
    "type": "Grow",
    "logical_cluster_name": "elastic_group",
    "start_time": "2023-06-05 01:58:43",
    "end_time": "2023-06-05 02:11:50",
    "result": "success",
    "log": "Expand from outside success"
  }, {
    "type": "Create",
    "logical_cluster_name": "test_logical",
    "start_time": "2023-06-21 08:35:58",
    "end_time": "2023-06-21 08:36:14",
    "result": "failed",
```

```
"log" : "list index out of range\\nChecking whether the reentry command is consistent with the previous command."
  } ],
  "count" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListLogicalClusterTasksSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListLogicalClusterTasksRequest request = new ListLogicalClusterTasksRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        request.withLogicalClusterName("<logical_cluster_name>");
        request.withType("<type>");
        request.withOrderBy("<order_by>");
        request.withOrder("<order>");
        try {
            ListLogicalClusterTasksResponse response = client.listLogicalClusterTasks(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListLogicalClusterTasksRequest()
        request.offset = <offset>
        request.limit = <limit>
        request.logical_cluster_name = "<logical_cluster_name>"
        request.type = "<type>"
        request.order_by = "<order_by>"
        request.order = "<order>"
        response = client.list_logical_cluster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.ListLogicalClusterTasksRequest{}
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    logicalClusterNameRequest:= "<logical_cluster_name>"
    request.LogicalClusterName = &logicalClusterNameRequest
    typeRequest:= "<type>"
    request.Type = &typeRequest
    orderByRequest:= "<order_by>"
    request.OrderBy = &orderByRequest
    orderRequest:= "<order>"
    request.Order = &orderRequest
    response, err := client.ListLogicalClusterTasks(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	查询任务信息。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.8 查询逻辑集群磁盘信息

功能介绍

该接口用于查询逻辑集群磁盘信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/logical-clusters/volumes

表 5-47 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-48 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页查询，偏移。
limit	否	Integer	分页查询，每页显示的条目数量，默认值为10。

请求参数

无

响应参数

状态码： 200

表 5-49 响应 Body 参数

参数	参数类型	描述
volumes	Array of LogicalClusterVolume objects	逻辑集群磁盘信息列表。
count	Long	逻辑集群磁盘总数。

表 5-50 LogicalClusterVolume

参数	参数类型	描述
logical_cluster_name	String	逻辑集群名称。
usage	String	磁盘使用量。

参数	参数类型	描述
total	String	磁盘总量。
percent	String	磁盘使用比例。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters/volumes
```

响应示例

状态码： 200

查询集群磁盘信息成功。

```
{
  "volumes": [ {
    "logical_cluster_name": "v3_logical",
    "usage": "1.0G",
    "total": "10.0G",
    "percent": 0.1
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListLogicalClusterVolumesSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
        .build();
ListLogicalClusterVolumesRequest request = new ListLogicalClusterVolumesRequest();
request.withOffset(<offset>);
request.withLimit(<limit>);
try {
    ListLogicalClusterVolumesResponse response = client.listLogicalClusterVolumes(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListLogicalClusterVolumesRequest()
        request.offset = <offset>
        request.limit = <limit>
        response = client.list_logical_cluster_volumes(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListLogicalClusterVolumesRequest{}
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListLogicalClusterVolumes(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	查询集群磁盘信息。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.9 查询节点列表

功能介绍

该接口用于查询节点列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/nodes

表 5-51 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-52 Query 参数

参数	是否必选	参数类型	描述
deleted	否	String	保留，不推荐使用。可用 node_ids 代替： <ul style="list-style-type: none"> “deleted = true”：node_ids 非空。 “deleted = false”：node_ids 为空。
node_ids	否	Array	节点ID列表。
offset	否	Integer	分页查询，偏移量，支持范围：0~10000。
limit	否	Integer	分页查询，每页显示的条目数量，支持范围：1~1000，默认值为100。
filter_by	否	String	过滤字段： <ul style="list-style-type: none"> status：节点状态。 instCreateType：资源状态。

参数	是否必选	参数类型	描述
filter	否	String	过滤字段内容： <ul style="list-style-type: none"> ● status = ALL: 查询所有节点，包含已删除历史节点。 ● status = CREATING: 创建中。 ● status = FREE: 空闲节点。 ● status = INIT: 初始化。 ● status = ACTIVE: 可用。 ● status = FAILED: 不可用。 ● status = UNKNOWN: 未知。 ● status = CREATE_FAILED: 创建失败。 ● status = DELETING: 删除中。 ● status = DELETE_FAILED: 删除失败。 ● status = REBOOTING: 重启中。 ● status = STOPPED: 已停止。 ● instCreateType = ALL: 查询所有节点，包含已删除历史节点。 ● instCreateType = INST: 已使用。 ● instCreateType = NODE: 空闲。
order_by	否	String	排序字段： <ul style="list-style-type: none"> ● 节点ID: id。 ● 节点名称: name。 ● 节点状态: status。
order	否	String	排序方式： <ul style="list-style-type: none"> ● 升序: asc。 ● 降序: desc。

请求参数

无

响应参数

状态码： 200

表 5-53 响应 Body 参数

参数	参数类型	描述
node_list	Array of ClusterNodeInfo objects	逻辑集群节点列表。
count	Integer	逻辑集群节点总数。
failed_count	Integer	逻辑集群节点失败总数。

表 5-54 ClusterNodeInfo

参数	参数类型	描述
id	String	节点ID。
name	String	节点名称。
status	String	节点状态： <ul style="list-style-type: none"> ● CREATING：创建中。 ● FREE：空闲。 ● ACTIVE：可用。 ● FAILED：不可用。 ● UNKNOWN：未知。 ● SERVER_FAILED：内部错误。 ● CREATE_FAILED：创建失败。 ● DELETING：删除中。 ● DELETE_FAILED：删除失败。 ● DELETED：已删除。 ● REBOOTING：重启中。 ● RESTORING：恢复中。 ● MODIFYING：修改中。 ● MODIFYING_PORT：绑定端口中。 ● INIT：初始化中。 ● STOPPED：已停止。

参数	参数类型	描述
sub_status	String	节点子状态： <ul style="list-style-type: none"> ● READY：节点ECS准备完成。 ● PREPARED：节点软件安装完成。 ● INITED：集群已创建。 ● CREATED：节点创建完成。
spec	String	节点规格。
inst_create_type	String	实例创建类型。
alias_name	String	节点别名。
az_code	String	可用区编码。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/nodes?deleted=false&node_ids&offset=0&limit=10&filter_by=instCreateType&filter=INST
```

响应示例

状态码： 200

查询集群节点信息成功。

```
{
  "node_list": [ {
    "id": "16413746-258e-4a3c-bea9-8496fdbefde3",
    "name": "ty-default--BGy6PUIN-K-dws-cn-cn-2-1",
    "status": "ACTIVE",
    "sub_status": "CREATED",
    "spec": "dws2.kc1.2xlarge",
    "inst_create_type": null,
    "alias_name": null,
    "az_code": "cn-north-4c"
  }, {
    "id": "46d8754b-3ed3-4b92-85b5-7479ef5b10ac",
    "name": "ty-default--BGy6PUIN-K-dws-cn-cn-1-1",
    "status": "ACTIVE",
    "sub_status": "CREATED",
    "spec": "dws2.kc1.2xlarge",
    "inst_create_type": null,
    "alias_name": null,
    "az_code": "cn-north-4c"
  } ],
  "count": 2,
  "failed_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class ListClusterNodesSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterNodesRequest request = new ListClusterNodesRequest();
        request.withDeleted("<deleted>");
        request.withNodeIds();
        request.withOffset("<offset>");
        request.withLimit("<limit>");
        request.withFilterBy("<filter_by>");
        request.withFilter("<filter>");
        request.withOrderBy("<order_by>");
        request.withOrder("<order>");
        try {
            ListClusterNodesResponse response = client.listClusterNodes(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterNodesRequest()
        request.deleted = "<deleted>"
        request.node_ids =
        request.offset = <offset>
        request.limit = <limit>
        request.filter_by = "<filter_by>"
        request.filter = "<filter>"
        request.order_by = "<order_by>"
        request.order = "<order>"
        response = client.list_cluster_nodes(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterNodesRequest{
        deletedRequest:= "<deleted>"
```

```

request.Deleted = &deletedRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
filterByRequest:= "<filter_by>"
request.FilterBy = &filterByRequest
filterRequest:= "<filter>"
request.Filter = &filterRequest
orderByRequest:= "<order_by>"
request.OrderBy = &orderByRequest
orderRequest:= "<order>"
request.Order = &orderRequest
response, err := client.ListClusterNodes(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	查询集群节点信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.10 查询逻辑集群可用 ring 环节点信息

功能介绍

该接口用于查询逻辑集群可用ring环节点信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/logical-clusters/rings

表 5-55 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-56 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页查询，偏移。
limit	否	Integer	分页查询，每页显示的条目数量，默认值为10。

请求参数

无

响应参数

状态码： 200

表 5-57 响应 Body 参数

参数	参数类型	描述
cluster_rings	Array of LogicalClusterRingInfo objects	集群环列表信息。
count	Integer	集群环数量。

表 5-58 LogicalClusterRingInfo

参数	参数类型	描述
ring_hosts	Array of RingHost objects	集群主机环信息。

表 5-59 RingHost

参数	参数类型	描述
host_name	String	主机名称。
back_ip	String	后端IP地址。
cpu_cores	Integer	主机CPU核数。
memory	Double	主机内存。
disk_size	Double	主机磁盘大小。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters/rings
```

响应示例

状态码： 200

查询逻辑集群可用ring环节点信息成功。

```
{
  "cluster_rings": [ {
    "ring_hosts": [ {
      "host_name": "host-172-16-20-246",
      "back_ip": "172.16.73.90",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }, {
      "host_name": "host-172-16-4-26",
      "back_ip": "172.16.123.5",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }, {
      "host_name": "host-172-16-43-90",
      "back_ip": "172.16.92.175",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }
  ]
}, {
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListLogicalClusterRingsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListLogicalClusterRingsRequest request = new ListLogicalClusterRingsRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ListLogicalClusterRingsResponse response = client.listLogicalClusterRings(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

```
try:
    request = ListLogicalClusterRingsRequest()
    request.offset = <offset>
    request.limit = <limit>
    response = client.list_logical_cluster_rings(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListLogicalClusterRingsRequest{}
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListLogicalClusterRings(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	查询逻辑集群可用ring环节点信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.11 查询集群拓扑 ring 环节点信息

功能介绍

该接口用于查询集群拓扑ring环节点信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/topo/rings

表 5-60 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-61 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页查询，偏移。
limit	否	Integer	分页查询，每页显示的条目数量，默认值为10。

请求参数

无

响应参数

状态码： 200

表 5-62 响应 Body 参数

参数	参数类型	描述
cluster_rings	Array of TopoRingInfo objects	集群拓扑环列表信息。
count	Integer	集群环数量。

表 5-63 TopoRingInfo

参数	参数类型	描述
instance_info_lists	Array of TopoInstanceInfo objects	集群实例列表信息。

表 5-64 TopoInstanceInfo

参数	参数类型	描述
id	String	实例ID。
name	String	实例名称。
manage_ip	String	实例管理IP。
traffic_ip	String	业务IP。
internal_ip	String	内部通信IP。
internal_mgmt_ip	String	内部管理IP。
eip	String	EIP。
elb	String	ELB地址。

参数	参数类型	描述
status	String	实例状态，有效值包括： <ul style="list-style-type: none"> ● 100：创建中。 ● 200：可用。 ● 300：不可用。 ● 303：创建失败。 ● 304：删除中。 ● 302：删除失败。 ● 400：已删除。 ● 910：停止中。 ● 900：已停止。 ● 920：启动中。 ● 800：已冻结。
az_code	String	可用区编码。

请求示例

```
GET https://{Endpoint}/v2/9b06d044ea4f49f1a58b2bed2b0084bd/clusters/9b7ff56b-47b3-4d00-a1fd-4c023d34404b/logical-clusters/rings
```

响应示例

状态码： 200

查询集群拓扑ring环节点信息成功。

```
{
  "cluster_rings": [ {
    "instance_info_lists": [ {
      "id": "a57e49db-c04b-45c7-9863-f7b6f3eed1b8",
      "name": "ty-default--BGy6PUIN-K-dws-cn-cn-1-1",
      "manage_ip": "172.16.26.233",
      "traffic_ip": "192.168.0.217",
      "internal_ip": "172.16.66.153",
      "internal_mgnt_ip": null,
      "eip": null,
      "elb": null,
      "status": 200,
      "az_code": "cn-north-4c"
    }, {
      "id": "3a37f794-be37-42d1-a299-a3eb94888ccb",
      "name": "ty-default--BGy6PUIN-K-dws-cn-cn-2-1",
      "manage_ip": "172.16.34.21",
      "traffic_ip": "192.168.0.80",
      "internal_ip": "172.16.65.89",
      "internal_mgnt_ip": null,
      "eip": null,
      "elb": null,
      "status": 200,
      "az_code": "cn-north-4c"
    }, {
      "id": "8763cbf1-5851-44a5-9e71-cbae35201f27",
      "name": "ty-default--BGy6PUIN-K-dws-dn-1-1",

```

```
"manage_ip" : "172.16.9.16",
"traffic_ip" : "192.168.0.88",
"internal_ip" : "172.16.67.64",
"internal_mgmt_ip" : null,
"eip" : null,
"elb" : null,
"status" : 200,
"az_code" : "cn-north-4c"
} ]
} ],
"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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListTopoRingsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTopoRingsRequest request = new ListTopoRingsRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ListTopoRingsResponse response = client.listTopoRings(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("<YOUR REGION>")) \  
        .build()   
  
    try:  
        request = ListTopoRingsRequest()  
        request.offset = <offset>  
        request.limit = <limit>  
        response = client.list_topo_rings(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"  
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
        dws.DwsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).  
Build()  
  
request := &model.ListTopoRingsRequest{  
offsetRequest:= int32(<offset>)  
request.Offset = &offsetRequest  
limitRequest:= int32(<limit>)  
request.Limit = &limitRequest  
response, err := client.ListTopoRings(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	查询集群拓扑ring环节点信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.12 查询集群详情

功能介绍

该接口用于查询集群详情。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/clusters/{cluster_id}
```

表 5-65 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	集群ID。获取方法，请参见 获取集群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 5-66 响应 Body 参数

参数	参数类型	描述
cluster	ClusterDetail object	集群详情对象。

表 5-67 ClusterDetail

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
status	String	<p>集群状态，有效值包括：</p> <ul style="list-style-type: none"> ● CREATING：创建中 ● ACTIVE：可用 ● FAILED：不可用 ● CREATE_FAILED：创建失败 ● DELETING：删除中 ● DELETE_FAILED：删除失败 ● DELETED：已删除 ● FROZEN：已冻结 <p>说明 如果用户查询返回状态为“AVAILABLE”，也表示可用。</p>
version	String	数据仓库版本。

参数	参数类型	描述
updated	String	集群上次修改时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ
created	String	集群创建时间，格式为ISO8601：YYYY-MM-DDThh:mm:ssZ
port	Integer	集群服务端口。
endpoints	Array of Endpoints objects	集群的内网连接信息。
nodes	Array of Nodes objects	集群实例。
tags	Array of Tags objects	集群标签。
user_name	String	管理员用户名。
number_of_node	Integer	集群节点数量。
recent_event	Integer	事件数。
availability_zone	String	可用区。
enterprise_project_id	String	企业项目ID。值为0表示默认企业项目“default”。
node_type	String	节点类型，详情请参见 数据仓库规格
vpc_id	String	虚拟私有云ID。
subnet_id	String	子网ID。
public_ip	PublicIp object	公网IP地址，如果未指定，则默认不使用公网连接。
public_endpoints	Array of PublicEndpoints objects	集群的公网连接信息，如果未指定，则默认不使用公网连接信息。

参数	参数类型	描述
action_progress	Map<String,String>	<p>Key值为正在进行的任务，有效值包括：</p> <ul style="list-style-type: none"> ● CREATING：创建中 ● RESTORING：恢复中 ● SNAPSHOTTING：快照中 ● GROWING：扩容中 ● REBOOTING：重启中 ● SETTING_CONFIGURATION：安全设置配置中 ● CONFIGURING_EXT_DATASOURCE：MRS连接配置中 ● ADD_CN_ING：增加CN中 ● DEL_CN_ING：删除CN中 ● REDISTRIBUTING：重分布中 ● ELB_BINDING：弹性负载均衡绑定中 ● ELB_UNBINDING：弹性负载均衡解绑中 ● ELB_SWITCHING：弹性负载均衡切换中 ● NETWORK_CONFIGURING：网络配置中 ● DISK_EXPANDING：磁盘扩容中 ● ACTIVE_STANDY_SWITCHOVER：主备恢复中 ● CLUSTER_SHRINKING：缩容中 ● SHRINK_CHECKING：缩容检测中 ● FLAVOR_RESIZING：规格变更中 ● MANAGE_IP_BINDING：登录开通中 ● FINE_GRAINED_RESTORING：细粒度恢复中 ● DR_RECOVERING：容灾恢复中 ● REPAIRING：修复中 <p>value值为正在进行任务的进度。</p>

参数	参数类型	描述
sub_status	String	<p>“可用”集群状态的子状态，有效值包括：</p> <ul style="list-style-type: none"> ● NORMAL：正常 ● READONLY：只读 ● REDISTRIBUTING：重分布中 ● REDISTRIBUTION-FAILURE：重分布失败 ● UNBALANCED：非均衡 ● UNBALANCED READONLY：非均衡，只读 ● DEGRADED：节点故障 ● DEGRADED READONLY：节点故障，只读 ● DEGRADED UNBALANCED：节点故障，非均衡 ● UNBALANCED REDISTRIBUTING：非均衡，重分布中 ● UNBALANCED REDISTRIBUTION-FAILURE：非均衡，重分布失败 ● READONLY REDISTRIBUTION-FAILURE：只读，重分布失败 ● UNBALANCED READONLY REDISTRIBUTION-FAILURE：非均衡，只读，重分布失败 ● DEGRADED REDISTRIBUTION-FAILURE：节点故障，重分布失败 ● DEGRADED UNBALANCED REDISTRIBUTION-FAILURE：节点故障，非均衡，只读，重分布失败 ● DEGRADED UNBALANCED READONLY REDISTRIBUTION-FAILURE：节点故障，非均衡，只读，重分布失败 ● DEGRADED UNBALANCED READONLY：节点故障，非均衡，只读 ● PENDING-REBOOT：待重启状态

参数	参数类型	描述
task_status	String	<p>集群管理任务，有效值包括：</p> <ul style="list-style-type: none"> ● UNFREEZING：解冻中 ● FREEZING：冻结中 ● RESTORING：恢复中 ● SNAPSHOTTING：快照中 ● GROWING：扩容中 ● REBOOTING：重启中 ● SETTING_CONFIGURATION：安全设置配置中 ● CONFIGURING_EXT_DATASOURCE：MRS连接配置中 ● DELETING_EXT_DATASOURCE：删除MRS连接 ● REBOOT_FAILURE：重启失败 ● RESIZE_FAILURE：扩容失败 ● ADD_CN_ING：增加CN中 ● DEL_CN_ING：删除CN中 ● CREATING_NODE：添加节点 ● CREATE_NODE_FAILED：添加节点失败 ● DELETING_NODE：删除节点 ● DELETE_NODE_FAILED：删除节点失败 ● REDISTRIBUTING：重分布中 ● REDISTRIBUTE_FAILURE：重分布失败 ● WAITING_REDISTRIBUTION：待重分布 ● REDISTRIBUTION_PAUSED：重分布暂停 ● ELB_BINDING：弹性负载均衡绑定中 ● ELB_BIND_FAILED：弹性负载均衡绑定失败 ● ELB_UNBINDING：弹性负载均衡解绑中 ● ELB_UNBIND_FAILED：弹性负载均衡解绑失败 ● ELB_SWITCHING：弹性负载均衡切换中 ● ELB_SWITCHING_FAILED：弹性负载均衡切换失败 ● NETWORK_CONFIGURING：网络配置中 ● NETWORK_CONFIG_FAILED：网络配置失败 ● DISK_EXPAND_FAILED：磁盘扩容失败 ● DISK_EXPANDING：磁盘扩容中 ● ACTIVE_STANDY_SWITCHOVER：主备恢复中

参数	参数类型	描述
		<ul style="list-style-type: none"> ACTIVE_STANDY_SWITCHOVER_FAILURE: 主备恢复失败 CLUSTER_SHRINK_FAILED: 缩容失败 CLUSTER_SHRINKING: 缩容中 SHRINK_CHECK_FAILED: 缩容检测失败 SHRINK_CHECKING: 缩容检测中 FLAVOR_RESIZING_FAILED: 规格变更失败 FLAVOR_RESIZING: 规格变更中 MANAGE_IP_BIND_FAILED: 登录开通失败 MANAGE_IP_BINDING: 登录开通中 ORDER_PENDING: 订单待支付 FINE_GRAINED_RESTORING: 细粒度恢复中 DR_RECOVERING: 容灾恢复中
parameter_group	ParameterGroup object	集群所关联的参数组信息。
node_type_id	String	节点类型ID。
security_group_id	String	安全组ID。
private_ip	Array of strings	内网IP地址列表。
maintain_window	MaintainWindow object	集群维护时间窗信息。
resize_info	ResizeInfo object	集群扩容状态详情。
failed_reasons	FailedReason object	失败原因。如果为空，则集群处于正常状态。
elb	ElbResp object	集群绑定的ELB信息。

表 5-68 Endpoints

参数	参数类型	描述
connect_info	String	内网连接信息。
jdbc_url	String	内网JDBC URL，默认格式如下： jdbc:postgresql://<connect_info>/<YOUR_DATABASE_NAME>

表 5-69 Nodes

参数	参数类型	描述
id	String	集群实例ID。
status	String	集群实例状态码。 <ul style="list-style-type: none"> • 100: 创建中 • 199: 空闲 • 200: 可用 • 300: 不可用 • 303: 创建失败 • 304: 删除中 • 305: 删除失败 • 400: 已删除

表 5-70 Tags

参数	参数类型	描述
key	String	键。输入标签键的最大长度为128个字符，不能为空字符串，且首尾字符不能为空格。不能包含“=”，“*”，“<”，“>”，“\”，“”，“ ”，“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。
value	String	值。输入标签值的最大长度为255个字符，首尾字符不能为空格，可以为空字符串。不能包含“=”，“*”，“<”，“>”，“\”，“”，“ ”，“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。

表 5-71 PublicIp

参数	参数类型	描述
public_bind_type	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> • auto_assign: 现在购买 • not_use: 暂未使用 • bind_existing: 使用已有
eip_id	String	弹性IP的id。

表 5-72 PublicEndpoints

参数	参数类型	描述
public_connect_info	String	公网连接信息。
jdbc_url	String	公网JDBC URL，默认格式如下： jdbc:postgresql://<public_connect_info>/<YOUR_DATABASE_name>

表 5-73 ParameterGroup

参数	参数类型	描述
id	String	参数组ID。
name	String	参数组名称。
status	String	集群参数状态，有效值包括： <ul style="list-style-type: none"> • In-Sync：已同步 • Applying：应用中 • Pending-Reboot：需重启生效 • Sync-Failure：应用失败

表 5-74 MaintainWindow

参数	参数类型	描述
day	String	每周的维护时间，以天为粒度，取值如下： <ul style="list-style-type: none"> • Mon：星期一 • Tue：星期二 • Wed：星期三 • Thu：星期四 • Fri：星期五 • Sat：星期六 • Sun：星期日
start_time	String	维护开始时间，显示格式为 HH:mm，时区为 GMT+0。
end_time	String	维护结束时间，显示格式为 HH:mm，时区为 GMT+0。

表 5-75 ResizeInfo

参数	参数类型	描述
target_node_number	Integer	扩容后的节点数量。
origin_node_number	Integer	扩容前的节点数量。
resize_status	String	扩容状态，取值如下： <ul style="list-style-type: none"> • GROWING：扩容中 • RESIZE_FAILURE：扩容失败
start_time	String	扩容开始时间，格式为ISO8601：YYYY-MM-DDThh:mm:ss

表 5-76 FailedReason

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

表 5-77 ElbResp

参数	参数类型	描述
public_ip	String	公网IP。
private_ip	String	内网IP。
private_endpoint	String	ELB终端地址。
name	String	ELB名称。
id	String	ELB的ID。
vpc_id	String	ELB所属VPC的ID。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7
```

响应示例

状态码： 200

查询集群详情成功。

```
{
  "cluster": {
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
    "name": "dws-1",
    "status": "AVAILABLE",
    "version": "1.2.0",
    "updated": "2018-02-10T14:28:14Z",
    "created": "2018-02-10T14:28:14Z",
    "port": 8000,
    "endpoints": [ {
      "connect_info": "dws-1.cn-north-4.myhuaweicloud.com",
      "jdbc_url": "jdbc:postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
    } ],
    "nodes": [ {
      "id": "acaf62a4-41b3-4106-bf6b-2f669d88291e",
      "status": "200"
    }, {
      "id": "d32de51e-4fcd-4e5a-a9dc-bb903abb494b",
      "status": "200"
    }, {
      "id": "d71a4a25-c9bc-4ffd-9f4a-e422aef327f9",
      "status": "200"
    } ],
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value2"
    } ],
    "user_name": "dbadmin",
    "number_of_node": 3,
    "recent_event": 6,
    "availability_zone": "cn-north-4b",
    "enterprise_project_id": "0",
    "node_type": "dws.m1.xlarge.ultrahigh",
    "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    "public_ip": {
      "public_bind_type": "auto_assign",
      "eip_id": "85b20d7e-9etypeb2a-98f3-3c8843ea3574"
    },
    "public_endpoints": [ {
      "public_connect_info": "dws-1.cn-north-4.myhuaweicloud.com",
      "jdbc_url": "jdbc:postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
    } ],
    "action_progress": {
      "SNAPSHOTTING": "20%"
    },
    "sub_status": "READONLY",
    "task_status": "SNAPSHOTTING",
    "parameter_group": {
      "id": "157e9cc4-64a8-11e8-adc0-fa7ae01bbebc",
      "name": "Default-Parameter-Group-dws ",
      "status": "In-Sync"
    },
    "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
    "private_ip": [ "192.168.0.12", "192.168.0.66" ],
    "maintain_window": {
      "day": "Wed",
      "start_time": "18:00",
      "end_time": "22:00"
    },
    "resize_info": {
      "target_node_num": "6",
      "origin_node_num": "3",
      "resize_status": "GROWING",
      "start_time": "2018-02-14T14:28:14Z"
    }
  }
}
```

```
}  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListClusterDetailsSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("cn-north-4"))  
            .build();  
        ListClusterDetailsRequest request = new ListClusterDetailsRequest();  
        try {  
            ListClusterDetailsResponse response = client.listClusterDetails(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListClusterDetailsRequest()
        response = client.list_cluster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterDetailsRequest{}
    response, err := client.ListClusterDetails(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	查询集群详情成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.13 查询节点类型

功能介绍

该接口用于查询所有GaussDB(DWS)服务支持的节点类型。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/node-types

表 5-78 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

无

响应参数

状态码： 200

表 5-79 响应 Body 参数

参数	参数类型	描述
node_types	Array of NodeTypes objects	节点类型对象列表。
count	Integer	节点类型总数。

表 5-80 NodeTypes

参数	参数类型	描述
spec_name	String	节点类型名称。
detail	Array of Detail objects	节点类型详细内容。
id	String	节点类型ID。
datastore_type	String	产品类型： <ul style="list-style-type: none"> dws: 标准数仓。 hybrid: 实时数仓。
architecture	String	架构类型： <ul style="list-style-type: none"> x86 arm
available_zones	Array of NodeTypeAvailableZones objects	支持的可用区及状态信息。
ram	Integer	内存大小。
vcpus	Integer	CPU数量。
datastores	Array of NodeTypeDatastores objects	内核版本信息。
volume	VolumeResp object	如果规格为固定存储容量规格，则该属性为规格典配的存储容量信息，如果为弹性存储规格，则该属性为null。
elastic_volume_specs	Array of NodeTypeElasticVolumeSpecs objects	如果规格为弹性容量规格，则该属性为规格典配的弹性容量信息，包括存储类型、最小容量、最大容量以及步长信息，如果为固定存储规格，则该属性为null。

表 5-81 Detail

参数	参数类型	描述
type	String	属性类型。
value	String	属性值。
unit	String	属性单位。

表 5-82 NodeTypeAvailableZones

参数	参数类型	描述
code	String	可用区ID。
status	String	节点类型可用状态。 <ul style="list-style-type: none"> • normal: 可用 • sellout: 售罄 • abandon: 不可用

表 5-83 NodeTypeDatastores

参数	参数类型	描述
version	String	内核版本号。
attachments	Attachments object	内核版本附加信息。

表 5-84 attachments

参数	参数类型	描述
min_cn	String	内核版本支持的最小CN。
max_cn	String	内核版本支持的最大CN。

表 5-85 VolumeResp

参数	参数类型	描述
type	String	磁盘类型，仅支持SSD。
size	Integer	磁盘可用容量。

表 5-86 NodeTypeElasticVolumeSpecs

参数	参数类型	描述
type	String	云盘存储类型。
step	String	云盘容量调整步长。
min_size	Integer	云盘支持的最小容量。
max_size	Integer	云盘支持的最大容量。

请求示例

GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/node-types

响应示例

状态码： 200

查询节点类型成功。

```
{
  "node_types": [ {
    "spec_name": "dws.d2.xlarge",
    "id": "ebe532d6-665f-40e6-a4d4-3c51545b6a67",
    "datastore_type": "dws",
    "architecture": "x86",
    "available_zones": [ {
      "code": "az1",
      "status": "normal"
    } ],
    "ram": 32,
    "vcpus": 4,
    "datastores": [ {
      "version": "8.1.3.200",
      "attachments": {
        "min_cn": 2,
        "max_cn": 20
      }
    } ],
    "volume": {
      "type": "SSD",
      "size": 160
    },
    "elastic_volume_specs": [ {
      "type": "SSD",
      "step": 100,
      "min_size": 100,
      "max_size": 32000
    } ],
    "detail": [ {
      "type": "vCPU",
      "value": "4"
    }, {
      "value": "1675",
      "type": "LOCAL_DISK",
      "unit": "GB"
    }, {
      "type": "mem",
      "value": "32",
      "unit": "GB"
    } ]
  } ],
}
```

```
"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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListNodeTypesSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("cn-north-4"))  
            .build();  
        ListNodeTypesRequest request = new ListNodeTypesRequest();  
        try {  
            ListNodeTypesResponse response = client.listNodeTypes(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListNodeTypesRequest()
        response = client.list_node_types(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListNodeTypesRequest{}
    response, err := client.ListNodeTypes(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	查询节点类型成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.14 删除集群

功能介绍

此接口用于删除集群。集群删除后将释放此集群的所有资源，包括用户数据。为了安全起见，请在删除集群前为这个集群创建快照。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v1.0/{project_id}/clusters/{cluster_id}
```

表 5-87 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待删除集群的ID。获取方法，请参见 获取集群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 5-88 请求 Body 参数

参数	是否必选	参数类型	描述
keep_last_manual_snapshot	是	Integer	指定集群需要保留的最近创建的手动快照。

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90
{
  "keep_last_manual_snapshot": 0
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteClusterSolution {

    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);
```

```
DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
DeleteClusterRequest request = new DeleteClusterRequest();
DeleteClusterRequestBody body = new DeleteClusterRequestBody();
body.withKeepLastManualSnapshot(0);
request.withBody(body);
try {
    DeleteClusterResponse response = client.deleteCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteClusterRequest()
        request.body = DeleteClusterRequestBody(
            keep_last_manual_snapshot=0
        )
        response = client.delete_cluster(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"
dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteClusterRequest{}
    request.Body = &model.DeleteClusterRequestBody{
        KeepLastManualSnapshot: int32(0),
    }
    response, err := client.DeleteCluster(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
202	删除集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.15 删除逻辑集群

功能介绍

此接口用于删除逻辑集群。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/clusters/{cluster_id}/logical-clusters/{logical_cluster_id}
```

表 5-89 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待删除集群的ID。获取方法，请参见 获取集群ID 。
project_id	是	String	项目ID。获取方法，请参见 获取项目ID 。
logical_cluster_id	是	String	指定待删除逻辑集群的ID。

请求参数

无

响应参数

状态码： 200

表 5-90 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
DELETE https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/logical-clusters/0b494d0d-8431-4c4f-8a06-2cc42d0d0c7d
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteLogicalClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteLogicalClusterRequest request = new DeleteLogicalClusterRequest();
        try {
            DeleteLogicalClusterResponse response = client.deleteLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = DeleteLogicalClusterRequest()
    response = client.delete_logical_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteLogicalClusterRequest{}
    response, err := client.DeleteLogicalCluster(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	删除逻辑集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.16 编辑逻辑集群

功能介绍

此接口用于编辑修改逻辑集群。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/logical-clusters/{logical_cluster_id}

表 5-91 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待编辑集群的ID。获取方法，请参见 获取集群ID 。
project_id	是	String	项目ID。获取方法，请参见 获取项目ID 。
logical_cluster_id	是	String	指定待编辑逻辑集群的ID。

请求参数

表 5-92 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_rings	是	Array of ClusterRing objects	逻辑集群编辑环列表信息。
mode	否	String	模式。
waiting_for_killing	否	Integer	是否等待销毁。

表 5-93 ClusterRing

参数	是否必选	参数类型	描述
ring_hosts	是	Array of RingHost objects	集群主机信息。

表 5-94 RingHost

参数	是否必选	参数类型	描述
host_name	是	String	主机名称。
back_ip	是	String	后端IP地址。
cpu_cores	是	Integer	主机CPU核数。
memory	是	Double	主机内存。
disk_size	是	Double	主机磁盘大小。

响应参数

状态码： 200

表 5-95 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/logical-clusters/0b494d0d-8431-4c4f-8a06-2cc42d0d0c7d

{
  "cluster_rings": [ {
    "ring_hosts": [ {
      "host_name": "host-172-16-20-246",
      "back_ip": "172.16.73.90",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }, {
      "host_name": "host-172-16-4-26",
      "back_ip": "172.16.123.5",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }, {
      "host_name": "host-172-16-43-90",
      "back_ip": "172.16.92.175",
      "cpu_cores": 8,
      "memory": 32.0,
      "disk_size": 800.0
    }
  ]
}, {
  "mode": null,
  "waiting_for_killing": 0
}
```

响应示例

状态码： 200

编辑逻辑集群成功。

```
{
  "error_code": "DWS.0000",
  "error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class UpdateLogicalClusterSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateLogicalClusterRequest request = new UpdateLogicalClusterRequest();
UpdateLogicalClusterRequestBody body = new UpdateLogicalClusterRequestBody();
List<RingHost> listClusterRingsRingHosts = new ArrayList<>();
listClusterRingsRingHosts.add(
    new RingHost()
        .withHostName("host-172-16-20-246")
        .withBackIp("172.16.73.90")
        .withCpuCores(8)
        .withMemory((double)32.0)
        .withDiskSize((double)800.0)
);
listClusterRingsRingHosts.add(
    new RingHost()
        .withHostName("host-172-16-4-26")
        .withBackIp("172.16.123.5")
        .withCpuCores(8)
        .withMemory((double)32.0)
        .withDiskSize((double)800.0)
);
listClusterRingsRingHosts.add(
    new RingHost()
        .withHostName("host-172-16-43-90")
        .withBackIp("172.16.92.175")
        .withCpuCores(8)
        .withMemory((double)32.0)
        .withDiskSize((double)800.0)
);
List<ClusterRing> listbodyClusterRings = new ArrayList<>();
listbodyClusterRings.add(
    new ClusterRing()
        .withRingHosts(listClusterRingsRingHosts)
);
body.withWaitingForKilling(0);
body.withClusterRings(listbodyClusterRings);
request.withBody(body);
try {
    UpdateLogicalClusterResponse response = client.updateLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateLogicalClusterRequest()
        listRingHostsClusterRings = [
            RingHost(
                host_name="host-172-16-20-246",
                back_ip="172.16.73.90",
                cpu_cores=8,
                memory=32.0,
                disk_size=800.0
            ),
            RingHost(
                host_name="host-172-16-4-26",
                back_ip="172.16.123.5",
                cpu_cores=8,
                memory=32.0,
                disk_size=800.0
            ),
            RingHost(
                host_name="host-172-16-43-90",
                back_ip="172.16.92.175",
                cpu_cores=8,
                memory=32.0,
                disk_size=800.0
            )
        ]
        listClusterRingsbody = [
            ClusterRing(
                ring_hosts=listRingHostsClusterRings
            )
        ]
        request.body = UpdateLogicalClusterRequestBody(
            waiting_for_killing=0,
            cluster_rings=listClusterRingsbody
        )
        response = client.update_logical_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateLogicalClusterRequest{}
    var listRingHostsClusterRings = []model.RingHost{
        {
            HostName: "host-172-16-20-246",
            BackIp: "172.16.73.90",
            CpuCores: int32(8),
            Memory: float64(32.0),
            DiskSize: float64(800.0),
        },
        {
            HostName: "host-172-16-4-26",
            BackIp: "172.16.123.5",
            CpuCores: int32(8),
            Memory: float64(32.0),
            DiskSize: float64(800.0),
        },
        {
            HostName: "host-172-16-43-90",
            BackIp: "172.16.92.175",
            CpuCores: int32(8),
            Memory: float64(32.0),
            DiskSize: float64(800.0),
        },
    }
    var listClusterRingsbody = []model.ClusterRing{
        {
            RingHosts: listRingHostsClusterRings,
        },
    }
    waitingForKillingUpdateLogicalClusterRequestBody := int32(0)
    request.Body = &model.UpdateLogicalClusterRequestBody{
        WaitingForKilling: &waitingForKillingUpdateLogicalClusterRequestBody,
        ClusterRings: listClusterRingsbody,
    }
    response, err := client.UpdateLogicalCluster(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	编辑逻辑集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.17 重启逻辑集群

功能介绍

该接口用于重启逻辑集群。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/clusters/{cluster_id}/logical-clusters/{logical_cluster_id}/restart
```

表 5-96 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定重启集群的ID。获取方法，请参见 获取集群ID 。
project_id	是	String	项目ID。获取方法，请参见 获取项目ID 。
logical_cluster_id	是	String	指定待重启逻辑集群的ID。

请求参数

无

响应参数

状态码： 200

表 5-97 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/logical-clusters/0b494d0d-8431-4c4f-8a06-2cc42d0d0c7d/restart
```

响应示例

状态码： 200

重启集群成功。

```
{  
  "error_code" : "DWS.0000",  
  "error_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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class RestartLogicalClusterSolution {  
  
    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
RestartLogicalClusterRequest request = new RestartLogicalClusterRequest();
try {
    RestartLogicalClusterResponse response = client.restartLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = RestartLogicalClusterRequest()
        response = client.restart_logical_cluster(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"

```

```

dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartLogicalClusterRequest{}
    response, err := client.RestartLogicalCluster(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	重启集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.18 切换逻辑集群开关

功能介绍

该接口用于切换逻辑集群开关，仅用于控制逻辑集群相关功能模块是否在页面显示。在集群已经是逻辑集群的场景下，修改该接口无任何作用及影响。

调用方法

请参见[如何调用API](#)。

URI

```
PUT /v2/{project_id}/clusters/{cluster_id}/logical-clusters/enable
```

表 5-98 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定集群的ID。获取方法，请参见 获取集群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 5-99 请求 Body 参数

参数	是否必选	参数类型	描述
enable	是	Boolean	切换开关-true。

响应参数

状态码： 200

表 5-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/logical-clusters/enable
```

```
{
  "enable": true
}
```

响应示例

状态码： 200

切换集群开关成功。

```
{
  "error_code": "DWS.0000",
  "error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class EnableLogicalClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        EnableLogicalClusterRequest request = new EnableLogicalClusterRequest();
        EnableLogicalClusterRequestBody body = new EnableLogicalClusterRequestBody();
        body.withEnable(true);
        request.withBody(body);
        try {
            EnableLogicalClusterResponse response = client.enableLogicalCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = EnableLogicalClusterRequest()
        request.body = EnableLogicalClusterRequestBody(
            enable=True
        )
        response = client.enable_logical_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.EnableLogicalClusterRequest{}
request.Body = &model.EnableLogicalClusterRequestBody{
    Enable: true,
}
response, err := client.EnableLogicalCluster(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	切换集群开关成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.19 删除空闲节点

功能介绍

该接口用于删除空闲节点。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/nodes/delete

表 5-101 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待删除空闲节点集群的ID。获取方法，请参见 获取集群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 5-102 请求 Body 参数

参数	是否必选	参数类型	描述
node_list	是	Array of strings	集群节点ID列表。
operate_type	是	String	操作类型，clear/delete。

响应参数

状态码： 202

表 5-103 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/nodes/delete
{
  "node_list": [ "16413746-258e-4a3c-bea9-8496fdbefde3","8bab1676-9332-4edd-b341-c3b95f08c19e","91fbaf6e-6374-4600-8f1c-e7553bdb9330"],
  "operate_type": "delete"
}
```

响应示例

状态码： 202

删除集群节点成功。

```
{
  "error_code": "DWS.0000",
```

```
"error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class DeleteClusterNodesSolution {

    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);

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

        DeleteClusterNodesRequest request = new DeleteClusterNodesRequest();
        DeleteClusterNodesRequestBody body = new DeleteClusterNodesRequestBody();
        List listbodyNodeList = new ArrayList<>();
        listbodyNodeList.add("16413746-258e-4a3c-bea9-8496fdbefde3");
        body.withOperateType("delete");
        body.withNodeList(listbodyNodeList);
        request.withBody(body);
        try {
            DeleteClusterNodesResponse response = client.deleteClusterNodes(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteClusterNodesRequest()
        listNodeListbody = [
            "16413746-258e-4a3c-bea9-8496fdbefde3"
        ]
        request.body = DeleteClusterNodesRequestBody(
            operate_type="delete",
            node_list=listNodeListbody
        )
        response = client.delete_cluster_nodes(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
```

```
dws.DwsClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.DeleteClusterNodesRequest{}
var listNodeListbody = []string{
    "16413746-258e-4a3c-bea9-8496fdbefde3",
}
request.Body = &model.DeleteClusterNodesRequestBody{
    OperateType: "delete",
    NodeList: listNodeListbody,
}
response, err := client.DeleteClusterNodes(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
202	删除集群节点成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.20 从空闲节点扩容

功能介绍

该接口用于从空闲节点扩容。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/resize-with-existed-nodes

表 5-104 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定节点集群的ID。获取方法，请参见 获取集群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 5-105 请求 Body 参数

参数	是否必选	参数类型	描述
scale_out	是	ScaleOut object	扩容对象。
force_backup	否	Boolean	是否强制备份。
mode	否	String	扩容备份模式，默认离线read-only。
logical_cluster_name	否	String	逻辑集群名称。非逻辑集群模式下该字段不填，逻辑集群模式下默认elastic_group。
expand_with_existed_node	是	Boolean	是否是使用已添加的空闲节点进行扩容。
auto_redistribute	否	Boolean	扩容完成后是否自动启动重分布，默认true。
redis_conf	否	RedisConfReq object	重分布配置信息。

表 5-106 ScaleOut

参数	是否必选	参数类型	描述
count	是	Integer	扩容节点数。有空闲节点时扩容必须使用所有节点。
subnet_id	否	String	指定子网ID。

表 5-107 RedisConfReq

参数	是否必选	参数类型	描述
redis_mode	是	String	重分布模式。offLine和onLine，默认offLine。
parallel_jobs	是	Integer	并行作业数量。可配置并发数在1-200之间，默认值：4。

响应参数

无

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/resize-with-existed-nodes
```

```
{
  "scale_out" : {
    "count" : 3
  },
  "expand_with_existed_node" : true,
  "auto_redistribute" : true,
  "redis_conf" : {
    "redis_mode" : "offLine",
    "parallel_jobs" : 4
  }
}
```

响应示例

状态码： 200

从空闲节点扩容成功。

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ResizeClusterWithExistedNodesSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ResizeClusterWithExistedNodesRequest request = new ResizeClusterWithExistedNodesRequest();
ResizeClusterWithExistedNodesRequestBody body = new ResizeClusterWithExistedNodesRequestBody();
BuildTaskInfo buildTaskInfo = new BuildTaskInfo();
buildTaskInfo.withBuildMode("read-only");
RedisConf redisConf = new RedisConf();
redisConf.withRedisMode("offLine")
    .withParallelJobs(4)
    .withParallelJob(4);
ScaleOut scaleOut = new ScaleOut();
scaleOut.withCount(3);
body.withBuildTaskInfo(buildTaskInfo);
body.withRedisConf(redisConf);
body.withIsSchedulerBuildMode(false);
body.withAutoRedistribute(true);
body.withCreateNodeOnly(true);
body.withExpandWithExistedNode(true);
body.withLogicalClusterName("test-logical-cluster");
body.withMode("read-only");
body.withForceBackup(false);
body.withScaleOut(scaleOut);
request.withBody(body);
try {
    ResizeClusterWithExistedNodesResponse response = client.resizeClusterWithExistedNodes(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```



```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = ResizeClusterWithExistedNodesRequest()
    buildTaskInfobody = BuildTaskInfo(
        build_mode="read-only"
    )
    redisConfbody = RedisConf(
        redis_mode="offLine",
        parallel_jobs=4,
        parallel_job=4
    )
    scaleOutbody = ScaleOut(
        count=3
    )
    request.body = ResizeClusterWithExistedNodesRequestBody(
        build_task_info=buildTaskInfobody,
        redis_conf=redisConfbody,
        is_scheduler_build_mode=False,
        auto_redistribute=True,
        create_node_only=True,
        expand_with_existed_node=True,
        logical_cluster_name="test-logical-cluster",
        mode="read-only",
        force_backup=False,
        scale_out=scaleOutbody
    )
    response = client.resize_cluster_with_existed_nodes(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.ResizeClusterWithExistedNodesRequest{}
    buildTaskInfobody := &model.BuildTaskInfo{
        BuildMode: "read-only",
    }
    redisConfbody := &model.RedisConf{
        RedisMode: "offLine",
        ParalleJobs: int32(4),
        ParalleJob: int32(4),
    }
    scaleOutbody := &model.ScaleOut{
        Count: int32(3),
    }
    request.Body = &model.ResizeClusterWithExistedNodesRequestBody{
        BuildTaskInfo: buildTaskInfobody,
        RedisConf: redisConfbody,
        IsSchedulerBuildMode: false,
        AutoRedistribute: true,
        CreateNodeOnly: true,
        ExpandWithExistedNode: true,
        LogicalClusterName: "test-logical-cluster",
        Mode: "read-only",
        ForceBackup: false,
        ScaleOut: scaleOutbody,
    }
    response, err := client.ResizeClusterWithExistedNodes(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	从空闲节点扩容成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.21 恢复重分布

功能介绍

此接口用于恢复暂停状态下的重分布操作，仅支持DWS2.0集群。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/clusters/{cluster_id}/redistribution/recovery
```

表 5-108 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定恢复重分布集群的ID。获取方法，请参见 获取集群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

无

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/redistribution/recovery
```

响应示例

状态码： 200

恢复重分布成功。

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class RestoreRedistributionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        RestoreRedistributionRequest request = new RestoreRedistributionRequest();
        try {
            RestoreRedistributionResponse response = client.restoreRedistribution(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

```
try:
    request = RestoreRedistributionRequest()
    response = client.restore_redistribution(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestoreRedistributionRequest{}
    response, err := client.RestoreRedistribution(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	恢复重分布成功。
400	请求错误。
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.22 暂停重分布

功能介绍

该接口用于暂停运行状态下的重分布操作，重分布暂停状态可设置重分布优先级，修改重分布并发数等操作。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/redistribution/suspend

表 5-109 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定暂停重分布集群的ID。获取方法，请参见 获取集群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

无

请求示例

POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/redistribution/suspend

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class StopRedistributionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        StopRedistributionRequest request = new StopRedistributionRequest();
        try {
            StopRedistributionResponse response = client.stopRedistribution(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = StopRedistributionRequest()
    response = client.stop_redistribution(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopRedistributionRequest{}
    response, err := client.StopRedistribution(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	暂停重分布成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.23 重启集群

功能介绍

此接口用于重启集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/restart

表 5-110 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待重启集群的ID。获取方法，请参见 获取集群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 5-111 请求 Body 参数

参数	是否必选	参数类型	描述
restart	是	Object	重启标识。

响应参数

无

请求示例

重启ID为“4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90”的集群:

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/restart

{
  "restart": { }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class RestartClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        RestartClusterRequest request = new RestartClusterRequest();
        RestartClusterRequestBody body = new RestartClusterRequestBody();
        body.withRestart(new Object());
        request.withBody(body);
        try {
            RestartClusterResponse response = client.restartCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = RestartClusterRequest()
        request.body = RestartClusterRequestBody(
            restart={}
        )
        response = client.restart_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.RestartClusterRequest{}
var restartRestartClusterRequestBody interface{} = make(map[string]string)
request.Body = &model.RestartClusterRequestBody{
    Restart: &restartRestartClusterRequestBody,
}
response, err := client.RestartCluster(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	重启集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.24 扩容集群

功能介绍

此接口用于扩容集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/resize

表 5-112 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待扩容集群的ID。获取方法，请参见 获取集群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 5-113 请求 Body 参数

参数	是否必选	参数类型	描述
scale_out	否	ScaleOut object	扩容对象。

表 5-114 ScaleOut

参数	是否必选	参数类型	描述
count	是	Integer	扩容节点数。
subnet_id	否	String	跨子网扩容指定的子网ID。不填默认为原集群的子网。

响应参数

无

请求示例

扩容集群新增3节点:

POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/resize

```
{
  "scale_out": {
    "count": 3
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ResizeClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ResizeClusterRequest request = new ResizeClusterRequest();
        ResizeClusterRequestBody body = new ResizeClusterRequestBody();
        ScaleOut scaleOutbody = new ScaleOut();
        scaleOutbody.withCount(3);
        body.withScaleOut(scaleOutbody);
        request.withBody(body);
        try {
            ResizeClusterResponse response = client.resizeCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ResizeClusterRequest()
        scaleOutbody = ScaleOut(
            count=3
        )
        request.body = ResizeClusterRequestBody(
            scale_out=scaleOutbody
        )
        response = client.resize_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ResizeClusterRequest{}
    scaleOutbody := &model.ScaleOut{
        Count: int32(3),
    }
}
```

```

request.Body = &model.ResizeClusterRequestBody{
    ScaleOut: scaleOutbody,
}
response, err := client.ResizeCluster(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	扩容集群调整集群大小成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.25 重置密码

功能介绍

此接口用于重置集群管理员密码。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/reset-password

表 5-115 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	指定待重置密码集群的ID。获取方法，请参见 获取集群ID 。

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 5-116 请求 Body 参数

参数	是否必选	参数类型	描述
new_password	是	String	<p>GaussDB(DWS) 集群管理员新密码。新密码复杂度要求如下：</p> <ul style="list-style-type: none"> • 密码字符长度为12~32位。 • 不能与用户名或倒序的用户名相同。 • 至少包含以下4种类型中的3种： <ul style="list-style-type: none"> - 小写字母 - 大写字母 - 数字 - 特殊字符（~!?,,:;"'(){}[]/ <>@#%^&*+ \=- ）。 • 不能与历史密码相同。 • 不能为弱密码。

响应参数

无

请求示例

重置ID为“4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90”的集群密码为“NewPassw0rd!”：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/reset-password
{
  "new_password": "NewPassw0rd!"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.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");

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

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ResetPasswordRequest request = new ResetPasswordRequest();
        ResetPasswordRequestBody body = new ResetPasswordRequestBody();
        body.withNewPassword("NewPassw0rd!");
        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

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ResetPasswordRequest()
    request.body = ResetPasswordRequestBody(
        new_password="NewPasswOrd!"
    )
    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

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ResetPasswordRequest{}
    request.Body = &model.ResetPasswordRequestBody{
        NewPassword: "NewPasswOrd!",
    }
    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	重置密码成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.26 创建集群前检查

功能介绍

该接口用于创建集群前预检查。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/cluster-precheck
```

表 5-117 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 5-118 请求 Body 参数

参数	是否必选	参数类型	描述
cluster	是	ClusterCheckBody object	集群校验对象。

表 5-119 ClusterCheckBody

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	企业项目ID。
flavor	是	String	集群规格名称。
availability_zones	是	Array of strings	可用区列表。
num_node	是	Integer	实例节点个数。
security_group_id	否	String	集群安全组ID。
datastore_version	是	String	集群版本。
vpc_id	是	String	集群虚拟私有云ID。
subnet_id	是	String	集群子网ID。
public_ip	否	OpenPublicIp object	弹性IP对象。
cross_spec_restore	否	String	跨规格恢复。
volume	否	Volume object	磁盘。
old_cluster_hostname	否	String	旧主机名。
restore_point	否	RestorePoint object	恢复指针。
tag_list	否	Array of Tag objects	标签列表。
dss_pool_id	否	String	存储池ID。
db_port	否	String	数据库端口。
db_password	否	String	管理员密码。

参数	是否必选	参数类型	描述
db_name	否	String	管理员用户名。
num_cn	否	Integer	CN节点数量。
name	否	String	集群名称。

表 5-120 OpenPublicIp

参数	是否必选	参数类型	描述
public_bind_type	否	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> • auto_assign：现在购买 • not_use：暂未使用 • bind_existing：使用已有
eip_id	否	String	弹性IP的ID。

表 5-121 Volume

参数	是否必选	参数类型	描述
volume	是	String	磁盘名称，取值如下： <ul style="list-style-type: none"> • SSD（超高IO） • 高IO（SAS） • 普通IO（SATA）
capacity	否	Integer	磁盘容量。

表 5-122 RestorePoint

参数	是否必选	参数类型	描述
back_ref	否	String	快照ID。
restore_time	否	Long	恢复时间。
cluster_id	否	String	集群ID。

表 5-123 Tag

参数	是否必选	参数类型	描述
key	否	String	标签key。

参数	是否必选	参数类型	描述
value	否	String	标签值。

响应参数

无

请求示例

创建集群操作前检查是否具备创建条件，比如在cn-north-7c可用区下是否有dws.m3.xlarge规格、3节点、公网IP、安全组、虚拟私有云等创建条件：

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/cluster-precheck
```

```
{
  "cluster" : {
    "flavor" : "dws.m3.xlarge",
    "num_node" : 3,
    "availability_zones" : [ "cn-north-7c" ],
    "vpc_id" : "c9f1171e-dc90-4ae9-bf22-f9736983ce2d",
    "subnet_id" : "ad9ef23f-1d03-4e52-af80-69604645e109",
    "security_group_id" : "f54a4572-d5d2-40b6-823a-79049bd5b71c",
    "public_ip" : {
      "public_bind_type" : "bind_existing",
      "eip_id" : "1923e8b0-cd40-4469-8cb1-64546c7b530c"
    }
  },
  "datastore_version" : "8.2.0"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class CheckClusterSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
CheckClusterRequest request = new CheckClusterRequest();
ClusterCheckRequestBody body = new ClusterCheckRequestBody();
OpenPublicIp publicIpCluster = new OpenPublicIp();
publicIpCluster.withPublicBindType("bind_existing")
    .withEipId("1923e8b0-cd40-4469-8cb1-64546c7b530c");
List<String> listClusterAvailabilityZones = new ArrayList<>();
listClusterAvailabilityZones.add("cn-north-7c");
ClusterCheckBody clusterbody = new ClusterCheckBody();
clusterbody.withFlavor("dws.m3.xlarge")
    .withAvailabilityZones(listClusterAvailabilityZones)
    .withNumNode(3)
    .withSecurityGroupId("f54a4572-d5d2-40b6-823a-79049bd5b71c")
    .withDatastoreVersion("8.2.0")
    .withVpcId("c9f1171e-dc90-4ae9-bf22-f9736983ce2d")
    .withSubnetId("ad9ef23f-1d03-4e52-af80-69604645e109")
    .withPublicIp(publicIpCluster);
body.withCluster(clusterbody);
request.withBody(body);
try {
    CheckClusterResponse response = client.checkCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \
```



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

try:
    request = CheckClusterRequest()
    publicIpCluster = OpenPublicIp(
        public_bind_type="bind_existing",
        eip_id="1923e8b0-cd40-4469-8cb1-64546c7b530c"
    )
    listAvailabilityZonesCluster = [
        "cn-north-7c"
    ]
    clusterbody = ClusterCheckBody(
        flavor="dws.m3.xlarge",
        availability_zones=listAvailabilityZonesCluster,
        num_node=3,
        security_group_id="f54a4572-d5d2-40b6-823a-79049bd5b71c",
        datastore_version="8.2.0",
        vpc_id="c9f1171e-dc90-4ae9-bf22-f9736983ce2d",
        subnet_id="ad9ef23f-1d03-4e52-af80-69604645e109",
        public_ip=publicIpCluster
    )
    request.body = ClusterCheckRequestBody(
        cluster=clusterbody
    )
    response = client.check_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CheckClusterRequest{
        publicBindTypePublicIp: "bind_existing"
    }
```

```

eipIdPublicIp:= "1923e8b0-cd40-4469-8cb1-64546c7b530c"
publicIpCluster := &model.OpenPublicIp{
    PublicBindType: &publicBindTypePublicIp,
    EipId: &eipIdPublicIp,
}
var listAvailabilityZonesCluster = []string{
    "cn-north-7c",
}
securityGroupIdCluster:= "f54a4572-d5d2-40b6-823a-79049bd5b71c"
clusterbody := &model.ClusterCheckBody{
    Flavor: "dws.m3.xlarge",
    AvailabilityZones: listAvailabilityZonesCluster,
    NumNode: int32(3),
    SecurityGroupId: &securityGroupIdCluster,
    DatastoreVersion: "8.2.0",
    VpcId: "c9f1171e-dc90-4ae9-bf22-f9736983ce2d",
    SubnetId: "ad9ef23f-1d03-4e52-af80-69604645e109",
    PublicIp: publicIpCluster,
}
request.Body = &model.ClusterCheckRequestBody{
    Cluster: clusterbody,
}
response, err := client.CheckCluster(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	创建集群前预校验成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.27 集群缩容

功能介绍

该接口用于缩容集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/cluster-shrink

表 5-124 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-125 请求 Body 参数

参数	是否必选	参数类型	描述
shrink_number	是	Integer	缩容数。 <ul style="list-style-type: none"> 集群按照环的方式配置，比如3个或4个主机组成一个环，这些主机上的DN主节点、备节点和从节点都部署在这个环里，缩容的最小单元是一个环。 缩容数应按扩容顺序逆向选择，如：先扩容3个节点，再扩容4个节点，本次缩容可选择缩容4个节点，或缩容7个节点。
online	是	Boolean	在线缩容。
type	是	String	数据库类型。 <ul style="list-style-type: none"> dws：标准数仓 hybrid：实时数仓
retry	否	Boolean	重试。
force_backup	是	Boolean	执行备份。
need_agency	是	Boolean	是否需要委托。缩容长时间执行需要用户授权委托，未授权委托可能缩容失败。

响应参数

表 5-126 响应参数

参数	参数类型	描述
job_id	String	扩容的任务ID。

请求示例

离线扩容ID为“4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90”的存算一体集群3个节点，并执行备份，不执行委托：

```
POST https://{Endpoint} /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/cluster-shrink
{
  "shrink_number": 3,
  "online": false,
  "type": "dws",
  "force_backup": true,
  "need_agency": false
}
```

响应示例

状态码： 200

```
{
  "job_id": "89cd04f168b84af6be287f71730fdb4b"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShrinkClusterSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ShrinkClusterRequest request = new ShrinkClusterRequest();
ClusterShrinkReq body = new ClusterShrinkReq();
body.withNeedAgency(false);
body.withForceBackup(true);
body.withType("dws");
body.withOnline(false);
body.withShrinkNumber(3);
request.withBody(body);
try {
    ShrinkClusterResponse response = client.shrinkCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShrinkClusterRequest()
        request.body = ClusterShrinkReq(
            need_agency=False,
            force_backup=True,
            type="dws",
            online=False,
            shrink_number=3
        )
        response = client.shrink_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShrinkClusterRequest{}
    request.Body = &model.ClusterShrinkReq{
        NeedAgency: false,
        ForceBackup: true,
        Type: "dws",
        Online: false,
        ShrinkNumber: int32(3),
    }
    response, err := client.ShrinkCluster(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	缩容请求下发成功。
400	请求错误。

状态码	描述
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.28 下发重分布

功能介绍

该接口用于集群扩容后将老节点数据均匀分布到新扩节点的数据重分布操作，数据“重分布”后将大大提升业务响应速率。重分布功能仅DWS 2.0和8.1.1.200及以上集群版本支持。离线调度重分布模式在8.2.0及以上版本将不再支持。只有在扩容之后，集群任务信息为“待重分布”状态时才能手动使用“重分布”功能，其他时段该功能不可使用。在扩容阶段也可以选择重分布模式等高级配置。重分布队列的排序依据表的relpage大小进行，为确保relpage大小正确，建议在重分布之前对需要重分布的表执行analyze操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/redistribution

表 5-127 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-128 请求 Body 参数

参数	是否必选	参数类型	描述
redis_mode	是	String	重分布模式，其中包括： <ul style="list-style-type: none"> • online：在线模式 • offline：离线模式
parallel_jobs	是	Integer	重分布并发数。取值范围 1~32。

响应参数

无

请求示例

下发集群在线重分布任务，设置并发数为3：

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/redistribution
{
  "redis_mode":"online",
  "parallel_jobs":3
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ExecuteRedistributionClusterSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ExecuteRedistributionClusterRequest request = new ExecuteRedistributionClusterRequest();
RedistributionReq body = new RedistributionReq();
body.withParallelJobs(3);
body.withRedisMode("online");
request.withBody(body);
try {
    ExecuteRedistributionClusterResponse response = client.executeRedistributionCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ExecuteRedistributionClusterRequest()
        request.body = RedistributionReq(
            parallel_jobs=3,
            redis_mode="online"
        )
        response = client.execute_redistribution_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteRedistributionClusterRequest{
        request.Body = &model.RedistributionReq{
            ParallelJobs: int32(3),
            RedisMode: "online",
        }
    }
    response, err := client.ExecuteRedistributionCluster(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	重分布下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。

状态码	描述
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.29 查询重分布详情

功能介绍

该接口用于查看当前集群的重分布模式、重分布进度、数据表重分布详情等监控信息。查看重分布详情功能仅DWS 2.0和8.1.1.200及以上集群版本支持，其中数据表重分布进度详情仅DWS 2.0和8.2.1及以上集群版本支持。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/redistribution

表 5-129 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-130 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页查询，每页大小。
offset	否	Integer	分页查询，偏移。
db_name	否	String	数据库名称。
table_name	否	String	表名称。
type	否	String	类型。

请求参数

无

响应参数

状态码： 200

表 5-131 响应 Body 参数

参数	参数类型	描述
redis_info	RdsRedisInfo object	重分布信息。
schedule_mode	Boolean	调度模式。
pause_enable	Boolean	是否允许暂停。
recover_enable	Boolean	是否允许恢复。
retry_enable	Boolean	是否允许重试。
update_enable	Boolean	是否允许更新。
control_enable	Boolean	是否允许控制。

表 5-132 RdsRedisInfo

参数	参数类型	描述
id	String	ID。
cluster_id	String	集群ID。
start_time	String	开始时间。
end_time	String	结束时间。
status	String	状态。
redis_conf	RedisConf object	重分布配置。
redis_progress	RedisProgress object	重分布进度信息。
redis_table_detail	RedisTableDetail object	重分布表信息。

表 5-133 RedisConf

参数	参数类型	描述
redis_mode	String	重分布模式。
schedule_conf	ScheduleConf object	调度配置信息。
parallel_jobs	Integer	并行作业数量。
parallel_job	Integer	并行作业数量。

表 5-134 ScheduleConf

参数	参数类型	描述
schedule_start	String	调度开始时间。
schedule_end	String	调度结束时间。
schedule_type	String	调度类型。
schedule_date	Array of integers	调度日期。
schedule_time	Array of strings	调度时间列表。

表 5-135 RedisProgress

参数	参数类型	描述
bytes_done	Long	已完成字节数。
byte_left	Long	剩余字节数。
tables_done	Integer	完成表数量。
tables_left	Integer	剩余表数量。
table_progress	Integer	表重分布进度。
total_progress	Integer	总进度。
redis_rate	String	重分布比例。
estimated_time	String	预计时间。
completed	Boolean	是否已完成。
initialed	Boolean	是否完成初始化。

参数	参数类型	描述
fail_count	Integer	失败总数。
redistributing	Boolean	cm_ctl 结果。
status	String	状态。
pause_by_user	Boolean	是否用户暂停。

表 5-136 RedisTableDetail

参数	参数类型	描述
data	Array of RedisTable objects	具体数据。
total	Integer	总条数。

表 5-137 RedisTable

参数	参数类型	描述
table_name	String	表名。
id	Integer	表唯一id。
schema_name	String	schema名。
logical_cluster_name	String	逻辑集群名。
size	Long	表大小。
status	String	重分布状态： <ul style="list-style-type: none"> • i: 重分布中。 • y: 重分布完成。 • n: 未开始。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/redistribution

{
  "redis_mode": "online",
  "parallel_jobs": 3
}
```

响应示例

状态码： 200

查询重分布详情成功。

```
{
  "redis_info": {
    "id": "8e4cc873-577a-4ca0-8a8c-723ef84581a9",
    "cluster_id": "f2175f2e-8379-4158-b521-6a961a952eb3",
    "start_time": "2023-01-11 09:21",
    "end_time": "2023-01-12 06:47",
    "status": "SUCCESS",
    "redis_conf": {
      "redis_mode": "online",
      "schedule_conf": {
        "schedule_start": "2023/01/11 16:41:27",
        "schedule_end": "2023/01/11 17:41:27",
        "schedule_type": "timeRange",
        "schedule_date": null,
        "schedule_time": null
      },
      "parallel_jobs": 4,
      "parallel_job": 4
    },
    "redis_progress": {
      "bytes_done": 191746048,
      "byte_left": 0,
      "tables_done": 163,
      "tables_left": 0,
      "table_progress": 100,
      "total_progress": 100,
      "redis_rate": null,
      "estimated_time": null,
      "completed": true,
      "initialed": true,
      "fail_count": 0,
      "redistributing": true,
      "status": "SUCCESS",
      "pause_by_user": false
    },
    "redis_table_detail": null
  },
  "schedule_mode": true,
  "pause_enable": true,
  "recover_enable": true,
  "retry_enable": true,
  "update_enable": true,
  "control_enable": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class ShowClusterRedistributionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ShowClusterRedistributionRequest request = new ShowClusterRedistributionRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withDbName("<db_name>");
        request.withTableName("<table_name>");
        request.withType("<type>");
        try {
            ShowClusterRedistributionResponse response = client.showClusterRedistribution(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()
```



```
try:
    request = ShowClusterRedistributionRequest()
    request.limit = <limit>
    request.offset = <offset>
    request.db_name = "<db_name>"
    request.table_name = "<table_name>"
    request.type = "<type>"
    response = client.show_cluster_redistribution(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterRedistributionRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    dbNameRequest := "<db_name>"
    request.DbName = &dbNameRequest
    tableNameRequest := "<table_name>"
    request.TableName = &tableNameRequest
    typeRequest := "<type>"
    request.Type = &typeRequest
    response, err := client.ShowClusterRedistribution(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	查询重分布详情成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.30 查询专属分布式存储池列表

功能介绍

该接口用于获取专属分布式存储池列表，只包括用户开通的SSD专属资源池信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/dss-pools

表 5-138 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

无

响应参数

状态码： 200

表 5-139 响应 Body 参数

参数	参数类型	描述
pools	Array of DssPool objects	专属分布式存储池列表对象。
count	Integer	专属分布式存储池数量。

表 5-140 DssPool

参数	参数类型	描述
id	String	专属分布式存储池ID。
name	String	专属分布式存储池名称。
type	String	专属分布式存储池的存储类型。 <ul style="list-style-type: none"> SSD: 超高IO专属分布式存储池。
project_id	String	专属分布式存储池归属的项目ID。
capacity	Integer	申请的专属分布式存储容量, 单位TB。
availability_zone	String	专属分布式存储池所属可用区。
status	String	专属分布式存储池的状态。 <ul style="list-style-type: none"> available: 专属分布式存储池处于可用状态。 deploying: 专属分布式存储池处于正在部署的过程中, 不可使用。 extending: 专属分布式存储池处于正在扩容的过程中, 可使用。
created_at	String	专属分布式存储池的创建时间。 <ul style="list-style-type: none"> 时间格式: UTC YYYY-MM-DDTHH:MM:SS

请求示例

GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/dss-pools

响应示例

状态码: 200

查询专属分布式存储池列表成功。

```
{
  "pools": [ {
    "id": "c950ee97-587c-4f24-8a74-3367e3da570f",
    "name": "pool-1",
    "type": "SSD",
    "project_id": "63d910f2705a487ebe4e1c274748d9e1",
```

```
"capacity" : "1000",  
"availability_zone" : "AZ1",  
"status" : "available",  
"created_at" : "2014-12-18T15:57:56.299000"  
}],  
"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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListDssPoolsSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListDssPoolsRequest request = new ListDssPoolsRequest();  
        try {  
            ListDssPoolsResponse response = client.listDssPools(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
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDssPoolsRequest()
        response = client.list_dss_pools(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDssPoolsRequest{}
    response, err := client.ListDssPools(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	查询专属分布式存储池列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.31 查询资源统计信息列表

功能介绍

该接口用于查询当前可用资源数量，其中包括“可用集群和总集群（个）”、“可用节点和总节点（个）”、“总容量（GB）”。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/statistics
```

表 5-141 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

无

响应参数

状态码： 200

表 5-142 响应 Body 参数

参数	参数类型	描述
statistics	Array of Statistic objects	资源数量信息列表。

表 5-143 Statistic

参数	参数类型	描述
name	String	资源名称。 <ul style="list-style-type: none"> cluster.total: 总集群 (个)。 cluster.normal: 可用集群 (个)。 instance.total: 总节点 (个)。 instance.normal: 可用节点 (个)。 storage.total: 总容量 (GB)。
value	Double	资源数量值。
unit	String	资源数量单位。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/statistics
```

响应示例

状态码： 200

查询资源数量成功。

```
{
  "statistics": [ {
    "name": "cluster.total",
    "value": 3,
    "unit": null
  }, {
    "name": "cluster.normal",
    "value": 3,
    "unit": null
  }, {
    "name": "instance.total",
    "value": 9,
    "unit": null
  }, {
    "name": "instance.normal",
    "value": 9,
    "unit": null
  } ]
}
```

```
}, {  
  "name" : "storage.total",  
  "value" : 1800,  
  "unit" : "GB"  
}]  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListStatisticsSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListStatisticsRequest request = new ListStatisticsRequest();  
        try {  
            ListStatisticsResponse response = client.listStatistics(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListStatisticsRequest()
        response = client.list_statistics(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListStatisticsRequest{}
    response, err := client.ListStatistics(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	查询资源数量成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.32 主备恢复

功能介绍

该接口用于集群主备恢复。当集群状态为“非均衡”时会出现某些节点主实例增多，从而负载压力较大。这种情况下集群状态是正常的，但整体性能要低于均衡状态。可进行集群主备恢复操作将集群状态切换为“可用”状态。

说明

- 集群主备恢复仅8.1.1.202及以上版本支持。
- 集群主备恢复将会短暂中断业务，中断时间根据用户自身业务量所决定，建议用户在业务低峰期执行此操作。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1.0/{project_id}/clusters/{cluster_id}/switchover
```

表 5-144 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

无

请求示例

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/switchover
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class SwitchOverClusterSolution {

    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);
```

```
DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
SwitchOverClusterRequest request = new SwitchOverClusterRequest();
try {
    SwitchOverClusterResponse response = client.switchOverCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = SwitchOverClusterRequest()
        response = client.switch_over_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchOverClusterRequest{}
    response, err := client.SwitchOverCluster(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	主备恢复成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.33 解除只读

功能介绍

该接口用于当集群进入只读状态时，无法进行数据库相关操作，用户可以在管理控制台解除集群的只读状态。触发只读状态可能是由于磁盘使用率过高，因此需要对集群数据进行清理或扩容。

📖 说明

解除只读仅支持1.7.2及以上集群版本。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1.0/{project_id}/clusters/{cluster_id}/cancel-readonly
```

表 5-145 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

无

请求示例

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/cancel-readonly
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class CancelReadOnlyClusterSolution {
    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CancelReadOnlyClusterRequest request = new CancelReadOnlyClusterRequest();
        try {
            CancelReadOnlyClusterResponse response = client.cancelReadOnlyCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CancelReadOnlyClusterRequest()
        response = client.cancel_readonly_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CancelReadOnlyClusterRequest{}
    response, err := client.CancelReadOnlyCluster(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	解除只读成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。

状态码	描述
500	服务内部错误。
503	服务不可用。

5.2.34 磁盘扩容

功能介绍

随着客户业务的发展，磁盘空间往往最先出现资源瓶颈，在其他资源尚且充足的情况下，通过磁盘扩容可快速缓解存储资源瓶颈现象，操作过程中无需暂停业务，并且不会造成CPU、内存等资源浪费。

说明

- 磁盘扩容功能仅8.1.1.203及以上版本支持，并且创建集群规格需要为存算一体SSD云盘类型。
- 按需+折扣套餐包消费模式下，存储扩容后超出折扣套餐包部分将按需收费。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/expand-instance-storage

表 5-146 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-147 请求 Body 参数

参数	是否必选	参数类型	描述
new_size	是	Integer	磁盘扩容后单节点有效存储容量（GB / 节点）。 该容量必须大于当前单节点有效容量，小于等于集群规格支持的单节点最大容量，扩容容量为规格支持的步长倍数。集群规格配置详情可根据 查询节点类型 查询。

响应参数

无

请求示例

集群磁盘扩容，扩容后单节点有效存储容量为200GB：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/expand-instance-storage
{
  "new_size" : 200
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ExpandInstanceStorageSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ExpandInstanceStorageRequest request = new ExpandInstanceStorageRequest();
ExpandInstanceStorage body = new ExpandInstanceStorage();
body.withNewSize(200);
request.withBody(body);
try {
    ExpandInstanceStorageResponse response = client.expandInstanceStorage(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ExpandInstanceStorageRequest()
        request.body = ExpandInstanceStorage(
            new_size=200
        )
        response = client.expand_instance_storage(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExpandInstanceStorageRequest{
        request.Body = &model.ExpandInstanceStorage{
            NewSize: int32(200),
        }
    }
    response, err := client.ExpandInstanceStorage(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	磁盘扩容成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。

状态码	描述
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.35 修改运维时间窗

功能介绍

您可以根据业务需求，设置可维护时间段。建议将可维护时间段设置在业务低峰期，避免业务在维护过程中异常中断。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1.0/{project_id}/clusters/{cluster_id}/maintenance-window

表 5-148 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-149 请求 Body 参数

参数	是否必选	参数类型	描述
day	是	String	日期，范围：Mon、Tue、Wed、Thu、Fri、Sat、Sun。
start_time	是	String	开始时间，UTC时间，格式为HH:mm，例如：22:00。 说明 <ul style="list-style-type: none"> 时间必须是整点。 开始时间和结束时间必须间隔4小时。

参数	是否必选	参数类型	描述
end_time	是	String	结束时间，UTC时间，格式为HH:mm，例如：02:00。 说明 <ul style="list-style-type: none"> 时间必须是整点。 开始时间和结束时间必须间隔4小时。

响应参数

无

请求示例

设置集群可维护时间段（每周四18:00~22:00）：

```
PUT https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/maintenance-window
```

```
{
  "day": "Thu",
  "start_time": "18:00",
  "end_time": "22:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateMaintenanceWindowSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateMaintenanceWindowRequest request = new UpdateMaintenanceWindowRequest();
MaintenanceWindow body = new MaintenanceWindow();
body.withEndTime("02:00");
body.withStartTime("22:00");
body.withDay("Thu");
request.withBody(body);
try {
    UpdateMaintenanceWindowResponse response = client.updateMaintenanceWindow(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateMaintenanceWindowRequest()
        request.body = MaintenanceWindow(
            end_time="02:00",
            start_time="22:00",
            day="Thu"
        )
        response = client.update_maintenance_window(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateMaintenanceWindowRequest{}
    request.Body = &model.MaintenanceWindow{
        EndTime: "02:00",
        StartTime: "22:00",
        Day: "Thu",
    }
    response, err := client.UpdateMaintenanceWindow(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	设置可维护时间段成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。

状态码	描述
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.36 查询集群 CN 节点

功能介绍

该接口用于查询指定集群的CN节点信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/cns

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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群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
min_num	Integer	允许的最小CN节点数量。
max_num	Integer	允许的最大CN节点数量。

参数	参数类型	描述
instances	Array of CoordinatorNode objects	CN节点列表。

表 5-152 CoordinatorNode

参数	参数类型	描述
id	String	节点ID。
name	String	节点名称。
private_ip	String	内网IP。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/cns
```

响应示例

状态码： 200

查询集群CN节点成功。

```
{
  "min_num" : 2,
  "max_num" : 3,
  "instances" : [ {
    "id" : "e07d1bfb-6eac-4827-96e0-d10b8ca29c41",
    "name" : "demo-dws-cn-cn-1-1",
    "private_ip" : "172.16.69.106"
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterCnSolution {
    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListClusterCnRequest request = new ListClusterCnRequest();
try {
    ListClusterCnResponse response = client.listClusterCn(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterCnRequest()
        response = client.list_cluster_cn(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterCnRequest{}
    response, err := client.ListClusterCn(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	查询集群CN节点成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.37 批量增加 CN 节点

功能介绍

该接口用于为指定集群批量增加CN节点。当用户集群创建后，实际需要的CN数量会随着业务需求而发生变化，因此管理CN节点功能的实现使用户可以根据实际需求动态调整集群CN数量。

说明

- 删除CN节点过程中不允许执行其他运维操作。
- 删除CN节点过程中需要停止业务操作，建议在业务低峰期或业务中断情况下进行操作。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1.0/{project_id}/clusters/{cluster_id}/cns/batch-create
```

表 5-153 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-154 请求 Body 参数

参数	是否必选	参数类型	描述
num	是	Integer	批量增加CN节点任务完成，集群总CN数量。 集群支持的CN节点数量与集群当前版本和节点数量相关，具体支持范围可根据 查询集群CN节点 查询，其中“min_num”为支持的最小数量，max_num为支持的最大数量。

响应参数

状态码： 200

表 5-155 响应 Body 参数

参数	参数类型	描述
job_id	String	批量增加CN节点任务ID。

请求示例

批量增加3CN节点：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/cns/batch-create
```

```
{
  "num" : 3
}
```

响应示例

状态码： 200

批量增加CN节点成功。

```
{
  "job_id" : "2c908185841339ce018414e9944b0020"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class BatchCreateClusterCnSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
        .build();
BatchCreateClusterCnRequest request = new BatchCreateClusterCnRequest();
BatchCreateCn body = new BatchCreateCn();
body.withNum(3);
request.withBody(body);
try {
    BatchCreateClusterCnResponse response = client.batchCreateClusterCn(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateClusterCnRequest()
        request.body = BatchCreateCn(
            num=3
        )
        response = client.batch_create_cluster_cn(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateClusterCnRequest{}
    request.Body = &model.BatchCreateCn{
        Num: int32(3),
    }
    response, err := client.BatchCreateClusterCn(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	批量增加CN节点成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.38 批量删除 CN 节点

功能介绍

该接口用于为指定集群删除CN节点。

当用户集群创建后，实际需要的CN数量会随着业务需求而发生变化，因此管理CN节点功能的实现使用户可以根据实际需求动态调整集群CN数量。

📖 说明

- 增删CN节点过程中不允许执行其他运维操作。
- 增删CN节点过程中需要停止业务操作，建议在业务低峰期或业务中断情况下进行操作。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1.0/{project_id}/clusters/{cluster_id}/cns/batch-delete
```

表 5-156 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-157 请求 Body 参数

参数	是否必选	参数类型	描述
instances	否	Array of strings	批量删除CN节点ID。

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	批量删除CN节点任务ID。

请求示例

批量删除ID为“b6ad3dc3-d2de-4d2c-a5df-fdde58eca8f0”的CN节点：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/cns/batch-delete

{
  "instances" : [ "b6ad3dc3-d2de-4d2c-a5df-fdde58eca8f0" ]
}
```

响应示例

状态码： 200

批量删除CN节点成功。

```
{
  "job_id" : "2c908185841339ce018414e9944b0020"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class BatchDeleteClusterCnSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteClusterCnRequest request = new BatchDeleteClusterCnRequest();
        BatchDeleteCn body = new BatchDeleteCn();
        List<String> listbodyInstances = new ArrayList<>();
```

```
listbodyInstances.add("b6ad3dc3-d2de-4d2c-a5df-fdde58eca8f0");
body.withInstances(listbodyInstances);
request.withBody(body);
try {
    BatchDeleteClusterCnResponse response = client.batchDeleteClusterCn(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchDeleteClusterCnRequest()
        listInstancesbody = [
            "b6ad3dc3-d2de-4d2c-a5df-fdde58eca8f0"
        ]
        request.body = BatchDeleteCn(
            instances=listInstancesbody
        )
        response = client.batch_delete_cluster_cn(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteClusterCnRequest{}
    var listInstancesbody = []string{
        "b6ad3dc3-d2de-4d2c-a5df-fdde58eca8f0",
    }
    request.Body = &model.BatchDeleteCn{
        Instances: &listInstancesbody,
    }
    response, err := client.BatchDeleteClusterCn(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CN节点成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.39 查询集群参数组

功能介绍

该接口用于查询集群所关联的参数组。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/configurations

表 5-159 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-160 响应 Body 参数

参数	参数类型	描述
configurations	Array of ClusterConfiguration objects	集群所关联的参数组信息。

表 5-161 ClusterConfiguration

参数	参数类型	描述
id	String	参数组ID。
name	String	参数组名称。
type	String	参数组类型。

参数	参数类型	描述
status	String	集群参数状态。 <ul style="list-style-type: none"> ● In-Sync: 已同步。 ● Applying: 应用中。 ● Pending-Reboot: 需重启生效。 ● Sync-Failure: 应用失败。
fail_reason	String	参数应用失败原因。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/configurations
```

响应示例

状态码： 200

查询集群参数组成功。

```
{
  "configurations": [ {
    "id": "509be37f-5420-40fb-89c2-0c5c9b8c90f6",
    "name": "defaultGUConfig",
    "type": "GUConfig",
    "status": "In-Sync",
    "fail_reason": "null"
  }, {
    "id": "3f62172a-8258-46a3-a761-e691265d6689",
    "name": "parameterGroupFor_3bc94408-265d-4076-8c6d-cc35ebc2c747",
    "type": "hiddenParameterGroup",
    "status": "In-Sync",
    "fail_reason": "null"
  }, {
    "id": "93cb2c86-a598-48ca-8c2c-0389f4b78511",
    "name": "defaultOBConfig",
    "type": "OBConfig",
    "status": "In-Sync",
    "fail_reason": "null"
  }, {
    "id": "3f377a36-1fa2-48fe-a1fa-a8801dae01ed",
    "name": "Default-Parameter-Template-DWS-8_2_0",
    "type": "parameterGroup",
    "status": "In-Sync",
    "fail_reason": "null"
  }, {
    "id": "d8f9ec5a-8235-4000-9433-187409738e8b",
    "name": "security-paragroup-dws",
    "type": "securityParameterGroup",
    "status": "In-Sync",
    "fail_reason": "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterConfigurationsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterConfigurationsRequest request = new ListClusterConfigurationsRequest();
        try {
            ListClusterConfigurationsResponse response = client.listClusterConfigurations(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListClusterConfigurationsRequest()
    response = client.list_cluster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterConfigurationsRequest{}
    response, err := client.ListClusterConfigurations(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	查询集群参数组成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.40 查询集群参数配置

功能介绍

该接口用于查询集群所关联的参数组。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/configurations/{configuration_id}

表 5-162 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。
configuration_id	是	String	参数组ID。获取方法，请参见 查询集群参数组 。

请求参数

无

响应参数

状态码： 200

表 5-163 响应 Body 参数

参数	参数类型	描述
configurations	Array of ConfigurationParameter objects	集群使用的参数配置信息。

表 5-164 ConfigurationParameter

参数	参数类型	描述
name	String	参数名称。
values	Array of ConfigurationParameterUnit objects	参数值。
unit	String	参数单位。
type	String	参数类型，包括： <ul style="list-style-type: none"> boolean string integer float list
readonly	Boolean	是否只读。
value_range	String	参数值范围。
restart_required	Boolean	是否需要重启。
description	String	参数描述。

表 5-165 ConfigurationParameterUnit

参数	参数类型	描述
type	String	参数类型，包括：cn、dn。
value	String	参数值。
default_value	String	参数默认值。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/configurations/2846d355-cd2a-4239-ae54-a7aef0b7963d
```

响应示例

状态码： 200

查询集群参数配置成功。

```
{
  "configurations" : [ {
    "name" : "enable_constraint_optimization",
    "values" : [ {
      "type" : "cn",
      "value" : "on",
      "default_value" : "on"
    }, {
      "type" : "dn",
      "value" : "on",
      "default_value" : "on"
    }
  ],
  "unit" : null,
  "type" : "boolean",
  "readonly" : false,
  "value_range" : "on|off",
  "restart_required" : false,
  "description" : "Specifies whether the informational constraint optimization execution plan can be used
for an HDFS foreign table."
} ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterConfigurationsParameterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
        .build();
    ListClusterConfigurationsParameterRequest request = new
ListClusterConfigurationsParameterRequest();
    try {
        ListClusterConfigurationsParameterResponse response =
client.listClusterConfigurationsParameter(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterConfigurationsParameterRequest()
        response = client.list_cluster_configurations_parameter(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterConfigurationsParameterRequest{}
    response, err := client.ListClusterConfigurationsParameter(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	查询集群参数配置成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.41 修改集群参数配置

功能介绍

该接口用于修改集群使用的参数配置信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/configurations/{configuration_id}

表 5-166 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。
configuration_id	是	String	参数组ID。

请求参数

表 5-167 请求 Body 参数

参数	是否必选	参数类型	描述
configurations	是	Array of ConfigurationParameterValue objects	集群参数配置列表。

表 5-168 ConfigurationParameterValue

参数	是否必选	参数类型	描述
type	是	String	参数类型，包括：cn、dn。
name	是	String	参数名称。
value	是	String	参数值。

响应参数

无

请求示例

集群中“enable_constraint_optimization”参数cn和dn类型修改为off:

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-
b4f3-0ad4e5e3ba90/configurations/2846d355-cd2a-4239-ae54-a7aef0b7963d

{
  "configurations" : [ {
    "type" : "cn",
    "name" : "enable_constraint_optimization",
    "value" : "off"
  }, {
    "type" : "dn",
    "name" : "enable_constraint_optimization",
    "value" : "off"
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class UpdateConfigurationSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateConfigurationRequest request = new UpdateConfigurationRequest();
        ConfigurationParameterValues body = new ConfigurationParameterValues();
        List<ConfigurationParameterValue> listbodyConfigurations = new ArrayList<>();
        listbodyConfigurations.add(
            new ConfigurationParameterValue()
                .withType("cn")
                .withName("enable_constraint_optimization")
                .withValue("off")
        );
    }
}
```

```
listbodyConfigurations.add(
    new ConfigurationParameterValue()
        .withType("dn")
        .withName("enable_constraint_optimization")
        .withValue("off")
);
body.withConfigurations(listbodyConfigurations);
request.withBody(body);
try {
    UpdateConfigurationResponse response = client.updateConfiguration(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateConfigurationRequest()
        listConfigurationsbody = [
            ConfigurationParameterValue(
                type="cn",
                name="enable_constraint_optimization",
                value="off"
            ),
            ConfigurationParameterValue(
                type="dn",
                name="enable_constraint_optimization",
                value="off"
            )
        ]
        request.body = ConfigurationParameterValues(
            configurations=listConfigurationsbody
        )
        response = client.update_configuration(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateConfigurationRequest{}
    var listConfigurationsbody = []model.ConfigurationParameterValue{
        {
            Type: "cn",
            Name: "enable_constraint_optimization",
            Value: "off",
        },
        {
            Type: "dn",
            Name: "enable_constraint_optimization",
            Value: "off",
        },
    }
    request.Body = &model.ConfigurationParameterValues{
        Configurations: listConfigurationsbody,
    }
    response, err := client.UpdateConfiguration(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	修改集群参数配置成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.42 快照统计信息

功能介绍

该接口用于快照统计信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/snapshots/statistics

表 5-169 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-170 响应 Body 参数

参数	参数类型	描述
statistics	Array of SnapshotsStatistic objects	快照统计信息。

表 5-171 SnapshotsStatistic

参数	参数类型	描述
name	String	资源统计信息名称。 <ul style="list-style-type: none"> storage.free: 免费容量。 storage.paid: 付费容量。 storage.used: 已用容量。
value	Number	资源统计信息值。
unit	String	资源统计信息单位。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/snapshots/statistics
```

响应示例

状态码： 200

查询快照统计信息成功。

```
{
  "statistics": [ {
    "name": "storage.free",
    "value": 300.0,
    "unit": "GB"
  }, {
    "name": "storage.paid",
    "value": 0,
    "unit": "GB"
  }, {
    "name": "storage.used",
    "value": 128.5,
    "unit": "GB"
  }
]
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotStatisticsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSnapshotStatisticsRequest request = new ListSnapshotStatisticsRequest();
        try {
            ListSnapshotStatisticsResponse response = client.listSnapshotStatistics(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DwsRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ListSnapshotStatisticsRequest()
    response = client.list_snapshot_statistics(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSnapshotStatisticsRequest{}
    response, err := client.ListSnapshotStatistics(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	查询快照统计信息成功。

状态码	描述
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.43 查询合适的缩容数

功能介绍

该接口用于查询合适的缩容数。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/shrink-numbers

表 5-172 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

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
shrink_sequence	Array of strings	合适的缩容数。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/shrink-numbers
```

响应示例

状态码： 200

```
{
  "shrink_sequence": [ 3, 7 ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterScaleInNumbersSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterScaleInNumbersRequest request = new ListClusterScaleInNumbersRequest();
        try {
            ListClusterScaleInNumbersResponse response = client.listClusterScaleInNumbers(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterScaleInNumbersRequest()
        response = client.list_cluster_scale_in_numbers(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListClusterScaleInNumbersRequest{}
response, err := client.ListClusterScaleInNumbers(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	查询合适的缩容数成功。
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.44 查询参数修改审计记录

功能介绍

该接口用于查询参数修改审计记录。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1/{project_id}/clusters/{cluster_id}/configurations/audit-records
```

表 5-174 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-175 Query 参数

参数	是否必选	参数类型	描述
action_time	否	Long	任务时间。
filter_by	否	String	过滤配置信息。
filter	否	String	过滤条件。
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

请求参数

无

响应参数

状态码： 200

表 5-176 响应 Body 参数

参数	参数类型	描述
records	Array of ConfigurationRecordResp objects	记录。
count	Integer	总数。

表 5-177 ConfigurationRecordResp

参数	参数类型	描述
id	String	ID。
operator	String	操作。

参数	参数类型	描述
start_time	String	开始时间。
end_time	String	结束时间。
status	String	状态。
failed_reason	String	失败原因。

请求示例

```
GET https://{Endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/configurations/audit-records
```

响应示例

状态码： 200

```
{
  "records" : [ {
    "id" : 1,
    "operator" : null,
    "start_time" : "2023-06-10T16:23:21.000Z",
    "end_time" : null,
    "status" : null,
    "failed_reason" : null
  }, {
    "id" : 2,
    "operator" : null,
    "start_time" : "2023-06-10T16:23:32.000Z",
    "end_time" : null,
    "status" : null,
    "failed_reason" : null
  } ],
  "count" : 10
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListConfigurationsAuditRecordsSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ListConfigurationsAuditRecordsRequest request = new ListConfigurationsAuditRecordsRequest();
request.withActionTime(<action_time>L);
request.withFilterBy("<filter_by>");
request.withFilter("<filter>");
request.withLimit(<limit>);
request.withOffset(<offset>);
try {
    ListConfigurationsAuditRecordsResponse response = client.listConfigurationsAuditRecords(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudskdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListConfigurationsAuditRecordsRequest()
        request.action_time = <action_time>
        request.filter_by = "<filter_by>"
        request.filter = "<filter>"
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_configurations_audit_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListConfigurationsAuditRecordsRequest{}
    actionTimeRequest := int64(<action_time>)
    request.ActionTime = &actionTimeRequest
    filterByRequest := "<filter_by>"
    request.FilterBy = &filterByRequest
    filterRequest := "<filter>"
    request.Filter = &filterRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListConfigurationsAuditRecords(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	参数修改审计记录查询成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.45 保存集群描述信息

功能介绍

该接口用于保存集群描述信息。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/description

表 5-178 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-179 Query 参数

参数	是否必选	参数类型	描述
namespace	否	String	命名空间。

请求参数

表 5-180 请求 Body 参数

参数	是否必选	参数类型	描述
description_info	是	String	集群描述信息。

响应参数

状态码： 200

表 5-181 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。

请求示例

```
POST https://{Endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/description
{
  "description_info": "desc info"
}
```

响应示例

状态码： 200

```
{
  "error_code": "DWS.0138",
  "error_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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class SaveClusterDescriptionInfoSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        SaveClusterDescriptionInfoRequest request = new SaveClusterDescriptionInfoRequest();
        request.withNamespace("<namespace>");
        ClusterDescriptionInfo body = new ClusterDescriptionInfo();
        body.withDescriptionInfo("desc info");
        request.withBody(body);
        try {
            SaveClusterDescriptionInfoResponse response = client.saveClusterDescriptionInfo(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = SaveClusterDescriptionInfoRequest()
```



```
request.namespace = "<namespace>"
request.body = ClusterDescriptionInfo(
    description_info="desc info"
)
response = client.save_cluster_description_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.SaveClusterDescriptionInfoRequest{}
    namespaceRequest := "<namespace>"
    request.Namespace = &namespaceRequest
    request.Body = &model.ClusterDescriptionInfo{
        DescriptionInfo: "desc info",
    }
    response, err := client.SaveClusterDescriptionInfo(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	集群描述信息保存成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.46 查询单个实例

功能介绍

该接口用于查询单个实例。

调用方法

请参见[如何调用API](#)。

URI

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

表 5-182 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 5-183 响应 Body 参数

参数	参数类型	描述
configuration_status	String	配置状态。
params_group_id	String	参数组ID。
type	String	类型。
subnet_id	String	子网ID。
role	String	角色。
internal_subnet_id	String	内部子网ID。
group	String	组。
secure_group	String	安全组。
vpc	String	VPC。
azcode	String	编码。
region	String	区域。
cluster_id	String	集群ID。
created	String	实例被创建的时间。
updated	String	实例被更新的时间。
status	String	状态。
name	String	名称。
links	Array of LinkResp objects	连接。
id	String	ID。
flavor	ClusterFlavorResp object	规格。
volume	CompatibleInstanceVolumeResp object	容量。
datastore	CompatibleDataStoreResp object	数据仓。
fault	CompatibleFaultResp object	是否当前规格。

参数	参数类型	描述
configuration	CompatibleConfigurationResp object	是否当前规格。
locality	String	地点。
replicas	Array of CompatibleReplicasResp objects	备份。
db_user	String	数据库用户。
storage_engine	String	存储引擎。
pay_model	Integer	付款方式。
public_ip	String	公网IP。
traffic_ip	String	流量IP。

表 5-184 ClusterFlavorResp

参数	参数类型	描述
id	String	ID。
links	Array of LinkResp objects	连接。

表 5-185 CompatibleInstanceVolumeResp

参数	参数类型	描述
type	String	类型。
used	Float	连接。
size	Integer	大小。

表 5-186 CompatibleDataStoreResp

参数	参数类型	描述
type	String	类型。
version	String	版本。

表 5-187 CompatibleFaultResp

参数	参数类型	描述
message	String	信息。
created	String	创建者。
details	String	详细。

表 5-188 CompatibleConfigurationResp

参数	参数类型	描述
id	String	ID。
name	String	名称。
links	Array of LinkResp objects	连接。

表 5-189 CompatibleReplicasResp

参数	参数类型	描述
id	String	ID。
name	String	名称。
links	Array of LinkResp objects	连接。

表 5-190 LinkResp

参数	参数类型	描述
rel	String	关联。
href	String	连接。

请求示例

```
GET https://{Endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/instances/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90
```

响应示例

状态码： 200

```
{
  "configuration_status" : "In-Sync",
  "params_group_id" : "53cb2c86-a598-58ca-8c2c-0389f4b78912",
  "type" : "dws-cn",
  "subnet_id" : "43cb2c86-a598-59ca-8c2c-0380f4b78613",
  "role" : "Standalone",
  "internal_subnet_id" : "73cb3c86-a598-59ca-8c2c-0380f4b78659",
  "group" : "cn-1",
  "secure_group" : null,
  "vpc" : "62cb2c86-a598-58ca-8c2c-0389f4b78954",
  "azcode" : "cn-north-854a",
  "region" : "cn-north-854",
  "cluster_id" : "85cb3c86-a598-59ca-8c2c-0380f4b78613",
  "created" : "2023-02-24T02:25:36",
  "updated" : "2023-02-24T09:25:36",
  "status" : null,
  "name" : "test-dws-cn-cn-1-1",
  "links" : [ {
    "rel" : "self",
    "href" : null
  }, {
    "rel" : "bookmark",
    "href" : null
  } ],
  "id" : "13cb2c86-a598-48ca-8c2c-0389f4b78963",
  "flavor" : {
    "id" : "46cb2c86-a598-48ca-8c2c-0389f4b78545",
    "links" : [ {
      "rel" : "self",
      "href" : null
    }, {
      "rel" : "bookmark",
      "href" : null
    } ]
  },
  "volume" : null,
  "datastore" : {
    "type" : "dws3.0",
    "version" : "9.0.0"
  },
  "fault" : null,
  "configuration" : [ {
    "id" : "56cb2c86-a598-48ca-8c2c-0389f4b78549",
    "name" : "test",
    "links" : [ {
      "rel" : "self",
      "href" : null
    }, {
      "rel" : "bookmark",
      "href" : null
    } ]
  } ],
  "locality" : null,
  "replicas" : null,
  "db_user" : "admin",
  "storage_engine" : null,
  "pay_model" : 0,
  "public_ip" : null,
  "traffic_ip" : "192.168.13.132"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowInstanceSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ShowInstanceRequest request = new ShowInstanceRequest();
        try {
            ShowInstanceResponse response = client.showInstance(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ShowInstanceRequest()
    response = client.show_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

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceRequest{}
    response, err := client.ShowInstance(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	查询单个实例成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.2.47 启动集群

功能介绍

该接口用于启动集群。已停止的集群想要恢复集群使用时，可通过启动集群功能触发集群启动，恢复集群业务。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/clusters/{cluster_id}/start
```

表 5-191 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-192 响应 Body 参数

参数	参数类型	描述
job_id	String	启动集群的任务ID。

请求示例

```
POST https://{endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/clusters/97cbaab3-939e-4dbc-9187-0fe240f2b9fd/start
```

响应示例

状态码： 200

启动集群下发成功

```
{
  "error_code" : null,
  "error_msg" : null,
  "job_id" : "2c9080d08cc99d28018ccd139e942498"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class StartClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        StartClusterRequest request = new StartClusterRequest();
```

```
request.withClusterId("{cluster_id}");
try {
    StartClusterResponse response = client.startCluster(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 huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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 = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = StartClusterRequest()
        request.cluster_id = "{cluster_id}"
        response = client.start_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.StartClusterRequest{}
request.ClusterId = "{cluster_id}"
response, err := client.StartCluster(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	启动集群下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	内部服务错误。
503	服务不可用。

5.2.48 停止集群

功能介绍

该接口用于停止集群。当用户的集群不再使用时，可通过停止集群功能来关闭集群，方便业务下线。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/clusters/{cluster_id}/stop
```

表 5-193 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-194 响应 Body 参数

参数	参数类型	描述
job_id	String	停止集群的任务ID。

请求示例

```
POST https://{endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/clusters/  
97cbaab3-939e-4dbc-9187-0fe240f2b9fd/stop
```

响应示例

状态码： 200

停止集群下发成功

```
{  
  "error_code" : null,  
  "error_msg" : null,  
  "job_id" : "2c9080d08cc99d28018ccd139e942498"  
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class StopClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        StopClusterRequest request = new StopClusterRequest();
        request.withClusterId("{cluster_id}");
        try {
            StopClusterResponse response = client.stopCluster(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 huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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 = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = StopClusterRequest()
    request.cluster_id = "{cluster_id}"
    response = client.stop_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopClusterRequest{}
    request.ClusterId = "{cluster_id}"
    response, err := client.StopCluster(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	停止集群下发成功。
400	请求错误。
401	鉴权失败。
500	内部服务错误。
503	服务不可用。

5.3 快照管理

5.3.1 创建快照

功能介绍

该接口用于为指定集群创建快照。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/snapshots

表 5-195 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 5-196 请求 Body 参数

参数	是否必选	参数类型	描述
snapshot	是	Snapshot object	快照对象。

表 5-197 Snapshot

参数	是否必选	参数类型	描述
name	是	String	快照名称，要求唯一性且必须以字母开头，不区分大小写，可以包含字母、数字、中划线或者下划线，不能包含其他的特殊字符，长度为4~64个字符。
cluster_id	是	String	指定创建快照的集群ID。获取方法，请参见 获取集群ID 。
description	否	String	快照描述，若不指定，描述为空。快照描述的字符长度不能超过256个字符，且不支持特殊字符!<>='&"。

响应参数

状态码： 200

表 5-198 响应 Body 参数

参数	参数类型	描述
snapshot	SnapshotRes p object	快照对象。

表 5-199 SnapshotResp

参数	参数类型	描述
id	String	快照ID。

请求示例

为ID为"44b277eb-39be-4921-be31-3d61b43651d7"的集群创建名为snapshot-3的手动快照：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots
{
  "snapshot" : {
    "name" : "snapshot-3",
    "cluster_id" : "44b277eb-39be-4921-be31-3d61b43651d7",
    "description" : "Snapshot-3 description"
  }
}
```

响应示例

状态码： 200

创建快照成功。

```
{
  "snapshot" : {
    "id" : "2a4d0f86-67cd-408a-8b66-017454fb7793"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateSnapshotSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CreateSnapshotRequest request = new CreateSnapshotRequest();
        CreateSnapshotRequestBody body = new CreateSnapshotRequestBody();
```

```
Snapshot snapshotbody = new Snapshot();
snapshotbody.setName("snapshot-3")
    .withClusterId("44b277eb-39be-4921-be31-3d61b43651d7")
    .withDescription("Snapshot-3 description");
body.withSnapshot(snapshotbody);
request.withBody(body);
try {
    CreateSnapshotResponse response = client.createSnapshot(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateSnapshotRequest()
        snapshotbody = Snapshot(
            name="snapshot-3",
            cluster_id="44b277eb-39be-4921-be31-3d61b43651d7",
            description="Snapshot-3 description"
        )
        request.body = CreateSnapshotRequestBody(
            snapshot=snapshotbody
        )
        response = client.create_snapshot(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateSnapshotRequest{}
    descriptionSnapshot := "Snapshot-3 description"
    snapshotbody := &model.Snapshot{
        Name: "snapshot-3",
        ClusterId: "44b277eb-39be-4921-be31-3d61b43651d7",
        Description: &descriptionSnapshot,
    }
    request.Body = &model.CreateSnapshotRequestBody{
        Snapshot: snapshotbody,
    }
    response, err := client.CreateSnapshot(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	创建快照成功。
400	请求错误。
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.2 查询快照列表

功能介绍

该接口用于查询快照列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/snapshots

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 。

请求参数

无

响应参数

状态码： 200

表 5-201 响应 Body 参数

参数	参数类型	描述
snapshots	Array of Snapshots objects	快照对象列表。
count	Integer	快照对象列表总数。

表 5-202 Snapshots

参数	参数类型	描述
id	String	快照ID。
name	String	快照名称。
description	String	快照描述。
started	String	快照创建的日期时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ。
finished	String	快照完成的日期时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ。
size	Double	快照大小，单位 GB。
status	String	快照状态： <ul style="list-style-type: none"> ● CREATING：创建中。 ● AVAILABLE：可用。 ● UNAVAILABLE：不可用。
type	String	快照创建类型。 <ul style="list-style-type: none"> ● MANUAL：手动快照。 ● AUTOMATED：自动快照。
cluster_id	String	快照对应的集群ID。

请求示例

GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots

响应示例

状态码： 200

查询快照列表成功。

```
{
  "snapshots": [ {
    "id": "2a4d0f86-67cd-408a-8b66-017454fb7793",
    "name": "snapshot-1",
    "description": "",
    "started": "2016-08-23T03:59:23Z",
    "finished": "2016-08-23T04:01:40Z",
    "size": 500,
    "status": "AVAILABLE",
    "type": "MANUAL",
    "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
  }, {
    "id": "4af11460-06ec-48a4-b3ad-0e3bbdcd8ab1",
    "name": "snapshot-2",
    "description": "",
    "started": "2016-08-23T18:20:00Z",
    "finished": "2016-08-23T18:22:12Z",
    "size": "500",
    "status": "AVAILABLE",
```

```
"type" : "MANUAL",
"cluster_id" : "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
}],
"count" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListSnapshotsRequest request = new ListSnapshotsRequest();
        try {
            ListSnapshotsResponse response = client.listSnapshots(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListSnapshotsRequest()
        response = client.list_snapshots(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListSnapshotsRequest{}
    response, err := client.ListSnapshots(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	查询快照列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.3 查询快照策略

功能介绍

该接口用于查询快照策略。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/snapshot-policies

表 5-203 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-204 响应 Body 参数

参数	参数类型	描述
keep_day	String	保留天数。
backup_strategies	Array of BackupStrategyDetail objects	备份策略。
device_name	String	备份设备。
server_ips	Array of strings	服务IP。
server_port	String	服务端口。
backup_param	String	备份参数。

表 5-205 BackupStrategyDetail

参数	参数类型	描述
policy_id	String	策略ID。
policy_name	String	策略名称。
backup_strategy	String	执行策略。
backup_type	String	备份类型： <ul style="list-style-type: none"> full：全量。 increment：增量。
backup_level	String	备份级别： <ul style="list-style-type: none"> cluster：集群级。
next_fire_time	String	下次触发时间。
update_time	String	更新时间。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/snapshot-policies
```

响应示例

```
{
  "keep_day": 3,
  "backup_strategies": [
    {
      "policy_id": "20c8ab14-9201-4d79-bb7c-5c42d1af84b1",
      "policy_name": "test_1031_02-default-full-strategy",
      "backup_strategy": "0 0 2 ? * 1",
      "backup_type": "full",
      "backup_level": "cluster",
      "next_fire_time": "1667700000000",
      "update_time": "2022-10-31T02:54:54"
    },
    {
      "policy_id": "a3c9e2fc-37aa-4088-b0d0-c49dccb0d2c2",
      "policy_name": "test_1031_02-default-increment-strategy",
      "backup_strategy": "0 56 2/8 ? * 2,3,4,5,6,7",
      "backup_type": "increment",
      "backup_level": "cluster",
      "next_fire_time": "1667213760000",
      "update_time": "2022-10-31T02:54:54"
    }
  ],
  "device_name": "obs",
  "server_ips": null,
  "server_port": null,
  "backup_param": ""
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotPolicySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSnapshotPolicyRequest request = new ListSnapshotPolicyRequest();
```

```

try {
    ListSnapshotPolicyResponse response = client.listSnapshotPolicy(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListSnapshotPolicyRequest()
        response = client.list_snapshot_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListSnapshotPolicyRequest{}
response, err := client.ListSnapshotPolicy(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	查询快照策略成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.4 查询快照详情

功能介绍

该接口用于使用快照ID查询快照详情。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/snapshots/{snapshot_id}

表 5-206 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 。
snapshot_id	是	String	快照ID。

请求参数

无

响应参数

状态码： 200

表 5-207 响应 Body 参数

参数	参数类型	描述
snapshot	SnapshotDetail object	快照详情对象。

表 5-208 SnapshotDetail

参数	参数类型	描述
id	String	快照ID。
name	String	快照名称。
description	String	快照描述。
started	String	快照创建的日期时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ。
finished	String	快照完成的日期时间，格式为 ISO8601：YYYY-MM-DDThh:mm:ssZ。
size	Double	快照大小，单位GB。
status	String	快照状态： <ul style="list-style-type: none"> ● CREATING：创建中。 ● AVAILABLE：可用。 ● UNAVAILABLE：不可用。

参数	参数类型	描述
type	String	快照创建类型。 <ul style="list-style-type: none"> MANUAL: 手动快照。 AUTOMATED: 自动快照。
cluster_id	String	快照对应的集群ID。
cluster_name	String	快照对应的集群名称。
backup_key	String	备份产生的Key。
prior_backup_key	String	增量快照使用前一个快照的BackupKey。当取值为FULL时表示这是一个全量快照。
base_backup_key	String	对应全量快照的BackupKey。
backup_device	String	备份介质。
total_backup_size	Long	快照累计大小, 表示从全量开始到当前增量的累计大小。单位KB
base_backup_name	String	对应全量快照名称。
support_inplace_restore	Boolean	是否支持恢复至原集群。
fine_grained_backup	Boolean	是否是细粒度备份。
fine_grained_backup_detail	FineGrainedSnapshotDetail object	细粒度备份信息。
guest_agent_version	String	集群guestAgent版本号。
cluster_status	String	集群状态 <ul style="list-style-type: none"> AVAILABLE: 可用 UNAVAILABLE: 不可用 FROZEN: 已冻结
bak_expected_start_time	String	预计快照开始时间。
bak_keep_day	Integer	备份保留天数。
bak_period	String	集群策略。
db_user	String	备份用户。
datastore	DataStore object	数据库版本

参数	参数类型	描述
progress	String	备份进度。
backup_level	String	备份级别 <ul style="list-style-type: none"> • cluster • schema

表 5-209 FineGrainedSnapshotDetail

参数	参数类型	描述
database	String	备份数据库。
schema_list	Array of String	备份模式集合。
table_list	Array of String	备份表集合。

表 5-210 DataStore

参数	参数类型	描述
type	String	数据库类型。
version	String	数据库版本。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/b5c45780-1006-49e3-b2d5-b3229975bbc7
```

响应示例

状态码： 200

查询快照详情成功。

```
{
  "snapshot" : {
    "id" : "2a4d0f86-67cd-408a-8b66-017454fb7793",
    "name" : "snapshot-1",
    "description" : "snapshot description",
    "started" : "2016-08-23T03:59:23Z",
    "finished" : "2016-08-23T04:01:40Z",
    "size" : 500,
    "status" : "AVAILABLE",
    "type" : "MANUAL",
    "cluster_id" : "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
  }
}
```


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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotDetailsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListSnapshotDetailsRequest request = new ListSnapshotDetailsRequest();
        try {
            ListSnapshotDetailsResponse response = client.listSnapshotDetails(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ListSnapshotDetailsRequest()
    response = client.list_snapshot_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListSnapshotDetailsRequest{}
    response, err := client.ListSnapshotDetails(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	查询快照详情成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.5 删除手动快照

功能介绍

该接口用于删除一个指定手动快照。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v1.0/{project_id}/snapshots/{snapshot_id}
```

表 5-211 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 。
snapshot_id	是	String	快照ID。

请求参数

无

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90
```

响应示例

```
status CODE 202
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteSnapshotSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        DeleteSnapshotRequest request = new DeleteSnapshotRequest();
        try {
            DeleteSnapshotResponse response = client.deleteSnapshot(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteSnapshotRequest()
        response = client.delete_snapshot(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSnapshotRequest{}
    response, err := client.DeleteSnapshot(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
202	删除快照成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.6 删除快照策略

功能介绍

该接口用于删除一个快照策略。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v1.0/{project_id}/clusters/{cluster_id}/snapshot-policies/{id}
```

表 5-212 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
id	是	String	快照策略ID。

请求参数

无

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/snapshot-policies/4db196c8-32fc-4bf1-bf57-05980bbda9a4
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteSnapshotPolicySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        DeleteSnapshotPolicyRequest request = new DeleteSnapshotPolicyRequest();
        try {
            DeleteSnapshotPolicyResponse response = client.deleteSnapshotPolicy(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("cn-north-4")) \  
        .build()  
  
    try:  
        request = DeleteSnapshotPolicyRequest()  
        response = client.delete_snapshot_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"  
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
        dws.DwsClientBuilder().  
            WithRegion(region.ValueOf("cn-north-4")).  
            WithCredential(auth).  
            Build())
```



```
request := &model.DeleteSnapshotPolicyRequest{}
response, err := client.DeleteSnapshotPolicy(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	删除快照策略成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.7 恢复快照到新集群

功能介绍

该接口用于使用快照恢复到新集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/snapshots/{snapshot_id}/actions

表 5-213 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 。
snapshot_id	是	String	待恢复的快照ID。

请求参数

表 5-214 请求 Body 参数

参数	是否必选	参数类型	描述
restore	是	Restore object	恢复对象。

表 5-215 Restore

参数	是否必选	参数类型	描述
name	是	String	集群名称，要求唯一性，必须以字母开头并只包含字母、数字、中划线，下划线，长度为4~64个字符且不超过64个字节，1个中文字符为3个字节，其他字符为1个字节。
subnet_id	否	String	指定子网ID，用于集群网络配置。默认值与原集群相同。
security_group_id	否	String	指定安全组ID，用于集群网络配置。默认值与原集群相同。
vpc_id	否	String	指定虚拟私有云ID，用于集群网络配置。默认值与原集群相同。
availability_zone	否	String	指定集群可用区。默认值与原集群相同。
port	否	Integer	指定集群服务端口，取值范围8000~30000，默认端口为8000。 最小值： 8000 最大值： 30000
public_ip	否	PublicIp object	公网IP地址，如果未指定，则默认不使用公网连接。
enterprise_project_id	否	String	企业项目ID，对集群指定企业项目，如果未指定，则使用默认企业项目“default”的ID，即0。
ipv6_enable	否	Boolean	指定网络协议类型。表明是否支持IPv6，默认不使用IPv6。

表 5-216 PublicIp

参数	是否必选	参数类型	描述
public_bind_type	否	String	弹性IP绑定类型，取值如下： <ul style="list-style-type: none"> • auto_assign：自动绑定。 • not_use：暂未使用。 • bind_existing：使用已有。
eip_id	否	String	弹性IP的ID。

响应参数

状态码： 200

表 5-217 响应 Body 参数

参数	参数类型	描述
cluster	Cluster object	集群对象。

表 5-218 Cluster

参数	参数类型	描述
id	String	集群ID。

请求示例

恢复快照到新集群dws-1，新集群在az1.dc1可用区下，数据库端口为8000，带子网ID，安全组ID，虚拟私有云ID，公网IP，企业ID等：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/actions
```

```
{
  "restore": {
    "name": "dws-1",
    "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
    "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    "availability_zone": "az1.dc1",
    "port": 8000,
    "public_ip": {
      "public_bind_type": "auto_assign",
      "eip_id": ""
    },
    "enterprise_project_id": "aca4e50a-266f-4786-827c-f8d6cc3fbada"
  }
}
```

响应示例

状态码： 200

恢复集群成功。

```
{
  "cluster" : {
    "id" : "7d85f602-a948-4a30-afd4-e84f47471c15"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class RestoreClusterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        RestoreClusterRequest request = new RestoreClusterRequest();
        RestoreClusterRequestBody body = new RestoreClusterRequestBody();
        PublicIp publicIpRestore = new PublicIp();
        publicIpRestore.withPublicBindType("auto_assign")
            .withEipId("");
        Restore restorebody = new Restore();
        restorebody.withName("dws-1")
            .withSubnetId("374eca02-cfc4-4de7-8ab5-dbebf7d9a720")
            .withSecurityGroupId("dc3ec145-9029-4b39-b5a3-ace5a01f772b")
            .withVpcId("85b20d7e-9eb7-4b2a-98f3-3c8843ea3574")
            .withAvailabilityZone("az1.dc1")
            .withPort(8000)
            .withPublicIp(publicIpRestore)
            .withEnterpriseProjectId("aca4e50a-266f-4786-827c-f8d6cc3fbada");
        body.withRestore(restorebody);
        request.withBody(body);
        try {
            RestoreClusterResponse response = client.restoreCluster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = RestoreClusterRequest()
        publicIpRestore = PublicIp(
            public_bind_type="auto_assign",
            eip_id=""
        )
        restorebody = Restore(
            name="dws-1",
            subnet_id="374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
            security_group_id="dc3ec145-9029-4b39-b5a3-ace5a01f772b",
            vpc_id="85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
            availability_zone="az1.dc1",
            port=8000,
            public_ip=publicIpRestore,
            enterprise_project_id="aca4e50a-266f-4786-827c-f8d6cc3fbada"
        )
        request.body = RestoreClusterRequestBody(
            restore=restorebody
        )
        response = client.restore_cluster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.RestoreClusterRequest{
        eipldPublicIp:= ""
        publicIpRestore := &model.PublicIp{
            PublicBindType: "auto_assign",
            Eipld: &eipldPublicIp,
        }
        subnetIdRestore:= "374eca02-cfc4-4de7-8ab5-dbebf7d9a720"
        securityGroupIdRestore:= "dc3ec145-9029-4b39-b5a3-ace5a01f772b"
        vpcIdRestore:= "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
        availabilityZoneRestore:= "az1.dc1"
        portRestore:= int32(8000)
        enterpriseProjectIdRestore:= "aca4e50a-266f-4786-827c-f8d6cc3fbada"
        restorebody := &model.Restore{
            Name: "dws-1",
            SubnetId: &subnetIdRestore,
            SecurityGroupId: &securityGroupIdRestore,
            VpcId: &vpcIdRestore,
            AvailabilityZone: &availabilityZoneRestore,
            Port: &portRestore,
            PublicIp: publicIpRestore,
            EnterpriseProjectId: &enterpriseProjectIdRestore,
        }
    }
    request.Body = &model.RestoreClusterRequestBody{
        Restore: restorebody,
    }
    response, err := client.RestoreCluster(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	恢复集群成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.8 用户恢复表名检测

功能介绍

该接口用于用户恢复表名检测。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/snapshots/{snapshot_id}/table-restore-check

表 5-219 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 。
snapshot_id	是	String	快照ID。

请求参数

表 5-220 请求 Body 参数

参数	是否必选	参数类型	描述
case_sensitive	是	Boolean	名称是否区分大小写。
database	是	String	数据库名称。
restore_table_list	是	Array of TableDetail objects	源表信息。
target_table_list	是	Array of TableDetail objects	目的表信息。

表 5-221 TableDetail

参数	是否必选	参数类型	描述
schema_name	是	String	schema名称。
table_name	是	String	表名称。

响应参数

状态码： 200

表 5-222 响应 Body 参数

参数	参数类型	描述
check_table_name_result	CheckTableNameResult object	检查结果。

表 5-223 CheckTableNameResult

参数	参数类型	描述
database	String	数据库名称。
restore_table_list	Array of strings	恢复源表信息。
target_table_list	Array of strings	恢复目的表信息。

请求示例

```
POST https://{Endpoint}/v2/0536cdee2200d5912f7cc00b877980f1/snapshots/c719b1a7-c85c-4cb5-a721-7694908c2c11/table-restore-check

{
  "case_sensitive" : true,
  "database" : "postgres",
  "restore_table_list" : [
    {
      "schema_name" : "postgres",
      "table_name" : "public"
    }
  ],
  "target_table_list" : [
    {
      "schema_name" : "postgres",
      "table_name" : "public"
    }
  ]
}
```

响应示例

状态码： 200

用户恢复表名检测成功。

```
{
  "check_table_name_result" : {
    "database" : "postgres",
    "restore_table_list" : null,
    "target_table_list" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class CheckTableRestoreSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
CheckTableRestoreRequest request = new CheckTableRestoreRequest();
CheckTableRestoreRequestBody body = new CheckTableRestoreRequestBody();
List<TableDetail> listbodyTargetTableList = new ArrayList<>();
listbodyTargetTableList.add(
    new TableDetail()
        .withSchemaName("postgres")
        .withTableName("public")
);
List<TableDetail> listbodyRestoreTableList = new ArrayList<>();
listbodyRestoreTableList.add(
    new TableDetail()
        .withSchemaName("postgres")
        .withTableName("public")
);
body.withTargetTableList(listbodyTargetTableList);
body.withRestoreTableList(listbodyRestoreTableList);
body.withDatabase("postgres");
body.withCaseSensitive(true);
request.withBody(body);
try {
    CheckTableRestoreResponse response = client.checkTableRestore(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()
```

```
try:
    request = CheckTableRestoreRequest()
    listTargetTableListbody = [
        TableDetail(
            schema_name="postgres",
            table_name="public"
        )
    ]
    listRestoreTableListbody = [
        TableDetail(
            schema_name="postgres",
            table_name="public"
        )
    ]
    request.body = CheckTableRestoreRequestBody(
        target_table_list=listTargetTableListbody,
        restore_table_list=listRestoreTableListbody,
        database="postgres",
        case_sensitive=True
    )
    response = client.check_table_restore(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CheckTableRestoreRequest{}
    var listTargetTableListbody = []model.TableDetail{
        {
            SchemaName: "postgres",
            TableName: "public",
        },
    }
    var listRestoreTableListbody = []model.TableDetail{
        {
```

```
        SchemaName: "postgres",
        TableName: "public",
    },
}
request.Body = &model.CheckTableRestoreRequestBody{
    TargetTableList: listTargetTableListbody,
    RestoreTableList: listRestoreTableListbody,
    Database: "postgres",
    CaseSensitive: true,
}
response, err := client.CheckTableRestore(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	用户恢复表名检测成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.9 恢复表

功能介绍

该接口用于恢复表。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/snapshots/{snapshot_id}/table-restore
```

表 5-224 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 。
snapshot_id	是	String	快照ID。

请求参数

表 5-225 请求 Body 参数

参数	是否必选	参数类型	描述
case_sensitive	是	Boolean	名称是否区分大小写。
database	是	String	数据库名称。
restore_table_list	是	Array of TableDetail objects	源表信息。
target_table_list	是	Array of TableDetail objects	目标表信息。

表 5-226 TableDetail

参数	是否必选	参数类型	描述
schema_name	是	String	schema名称。
table_name	是	String	表名称。

响应参数

状态码： 200

表 5-227 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。

请求示例

```
POST https://{Endpoint}/v2/0536cdee2200d5912f7cc00b877980f1/snapshots/c719b1a7-c85c-4cb5-a721-7694908c2c11/table-restore
```

```
{
  "case_sensitive" : true,
  "database" : "postgres",
  "restore_table_list" : [ {
    "schema_name" : "postgres",
    "table_name" : "public"
  } ],
  "target_table_list" : [ {
    "schema_name" : "postgres",
    "table_name" : "public"
  } ]
}
```

响应示例

状态码： 200

用户恢复表成功

```
{
  "job_id" : "2c9081c0894918c301894e503ef21b68"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class RestoreTableSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        RestoreTableRequest request = new RestoreTableRequest();
        RestoreTableRequestBody body = new RestoreTableRequestBody();
        List<TableDetail> listbodyTargetTableList = new ArrayList<>();
        listbodyTargetTableList.add(
```

```
        new TableDetail()
            .withSchemaName("postgres")
            .withTableName("public")
    );
    List<TableDetail> listbodyRestoreTableList = new ArrayList<>();
    listbodyRestoreTableList.add(
        new TableDetail()
            .withSchemaName("postgres")
            .withTableName("public")
    );
    body.withTargetTableList(listbodyTargetTableList);
    body.withRestoreTableList(listbodyRestoreTableList);
    body.withDatabase("postgres");
    body.withCaseSensitive(true);
    request.withBody(body);
    try {
        RestoreTableResponse response = client.restoreTable(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = RestoreTableRequest()
        listTargetTableListbody = [
            TableDetail(
                schema_name="postgres",
                table_name="public"
            )
        ]
        listRestoreTableListbody = [
            TableDetail(
                schema_name="postgres",
                table_name="public"
            )
        ]
```

```

    )
  ]
  request.body = RestoreTableRequestBody(
    target_table_list=listTargetTableListbody,
    restore_table_list=listRestoreTableListbody,
    database="postgres",
    case_sensitive=True
  )
  response = client.restore_table(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.RestoreTableRequest{}
    var listTargetTableListbody = []model.TableDetail{
        {
            SchemaName: "postgres",
            TableName: "public",
        },
    },
    var listRestoreTableListbody = []model.TableDetail{
        {
            SchemaName: "postgres",
            TableName: "public",
        },
    },
    request.Body = &model.RestoreTableRequestBody{
        TargetTableList: listTargetTableListbody,
        RestoreTableList: listRestoreTableListbody,
        Database: "postgres",
        CaseSensitive: true,
    }
    response, err := client.RestoreTable(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	用户恢复表成功
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.10 获取跨区域快照可用 Region

功能介绍

该接口用于获取跨区域快照可用Region。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1/{project_id}/snapshots/cross-regions
```

表 5-228 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 5-229 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页偏移，默认0。
limit	否	Integer	分页大小，默认10。

请求参数

无

响应参数

状态码： 200

表 5-230 响应 Body 参数

参数	参数类型	描述
regions	Array of SnapshotRegion objects	区域列表。
count	Integer	总数。

表 5-231 SnapshotRegion

参数	参数类型	描述
region_id	String	区域ID。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/snapshots/cross-regions
```

响应示例

状态码： 200

获取跨区域快照可用region成功：

```
{
  "regions": [ {
    "region_id": "cn-north-7"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotCrossRegionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListSnapshotCrossRegionRequest request = new ListSnapshotCrossRegionRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ListSnapshotCrossRegionResponse response = client.listSnapshotCrossRegion(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ListSnapshotCrossRegionRequest()
    request.offset = <offset>
    request.limit = <limit>
    response = client.list_snapshot_cross_region(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListSnapshotCrossRegionRequest{}
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListSnapshotCrossRegion(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	获取跨区域快照可用region成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.11 设置跨区域备份配置

功能介绍

该接口用于设置跨区域备份配置。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/snapshots/cross-region-policies
```

表 5-232 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 5-233 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

参数	是否必选	参数类型	描述
destination_project_id	是	String	目的项目ID。
destination_region	是	String	目的区域。
status	是	Boolean	状态。
back_keep_day	是	Integer	保留天数。

响应参数

无

请求示例

```
POST https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/snapshots/cross-region-policies
{
  "cluster_id": "4aed8622-e99c-4b60-bd35-1ccde7c26ad0",
  "destination_project_id": "xxx",
  "destination_region": "xxx",
  "status": true,
  "back_keep_day": 3
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteSnapshotCrossRegionPolicySolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
DeleteSnapshotCrossRegionPolicyRequest request = new DeleteSnapshotCrossRegionPolicyRequest();
request.withClusterId("<cluster_id>");
try {
    DeleteSnapshotCrossRegionPolicyResponse response =
client.deleteSnapshotCrossRegionPolicy(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteSnapshotCrossRegionPolicyRequest()
        request.cluster_id = "<cluster_id>"
        response = client.delete_snapshot_cross_region_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteSnapshotCrossRegionPolicyRequest{}
    request.ClusterId = "<cluster_id>"
    response, err := client.DeleteSnapshotCrossRegionPolicy(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	设置跨区域备份配置成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.12 查询所有跨区域快照配置

功能介绍

该接口用于查询所有跨区域快照配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/snapshots/cross-region-policies

表 5-234 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 5-235 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群ID。获取方法，请参见 获取集群ID 。
offset	否	Integer	分页偏移。
limit	否	Integer	分页大小。

请求参数

无

响应参数

状态码： 200

表 5-236 响应 Body 参数

参数	参数类型	描述
cross_region_configs	Array of CrossRegionSnapshotConfig objects	跨区域配置信息。
count	Integer	总数。

表 5-237 CrossRegionSnapshotConfig

参数	参数类型	描述
cluster_id	String	集群ID。
cluster_name	String	集群名称。
source_region	String	源区域。
source_project_id	String	源项目ID。
destination_region	String	目的区域。
destination_project_id	String	目的项目ID。
status	Boolean	状态： <ul style="list-style-type: none"> • true: 开启。 • false: 关闭。
back_keep_day	Integer	保存时间。
total_size	Long	总大小。

请求示例

GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/snapshots/cross-region-policies

响应示例

状态码： 200

查询所有跨区域快照配置成功：

```
{
  "cross_region_configs": [ {
    "cluster_id": "4aed8622-e99c-4b60-bd35-1ccde7c26ad0",
    "cluster_name": "dwstest-821-100",
    "source_region": null,
    "source_project_id": null,
    "destination_region": null,
    "destination_project_id": null,
    "status": null,
    "back_keep_day": 3,
    "total_size": null
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSnapshotCrossRegionPolicySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListSnapshotCrossRegionPolicyRequest request = new ListSnapshotCrossRegionPolicyRequest();
        request.withClusterId("<cluster_id>");
        request.withOffset("<offset>");
        request.withLimit("<limit>");
        try {
            ListSnapshotCrossRegionPolicyResponse response = client.listSnapshotCrossRegionPolicy(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ListSnapshotCrossRegionPolicyRequest()
    request.cluster_id = "<cluster_id>"
    request.offset = <offset>
    request.limit = <limit>
    response = client.list_snapshot_cross_region_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListSnapshotCrossRegionPolicyRequest{}
    clusterIdRequest := "<cluster_id>"
    request.ClusterId = &clusterIdRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListSnapshotCrossRegionPolicy(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	查询所有跨区域快照配置成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.13 删除跨区域备份配置

功能介绍

该接口用于删除跨区域备份配置。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/snapshots/cross-region-policies

表 5-238 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 5-239 Query 参数

参数	是否必选	参数类型	描述
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/snapshots/cross-region-policies?cluster=4aed8622-e99c-4b60-bd35-1ccde7c26ad0
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteSnapshotCrossRegionPolicySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        DeleteSnapshotCrossRegionPolicyRequest request = new DeleteSnapshotCrossRegionPolicyRequest();
        request.withClusterId("<cluster_id>");
        try {
            DeleteSnapshotCrossRegionPolicyResponse response =
            client.deleteSnapshotCrossRegionPolicy(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteSnapshotCrossRegionPolicyRequest()
        request.cluster_id = "<cluster_id>"
        response = client.delete_snapshot_cross_region_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.DeleteSnapshotCrossRegionPolicyRequest{}
request.ClusterId = "<cluster_id>"
response, err := client.DeleteSnapshotCrossRegionPolicy(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	删除跨区域备份配置成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.3.14 复制快照

功能介绍

该接口用于复制一个自动快照。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/snapshots/{snapshot_id}/linked-copy

表 5-240 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 。
snapshot_id	是	String	快照ID。

请求参数

表 5-241 请求 Body 参数

参数	是否必选	参数类型	描述
backup_name	是	String	快照名称，要求唯一性且必须以字母开头，不区分大小写，可以包含字母、数字、中划线或者下划线，不能包含其他的特殊字符，长度为4~64个字符。
description	否	String	快照描述，若不指定，描述为空。快照描述的字符长度不能超过256个字符，且不支持特殊字符!<>'=&"。

响应参数

表 5-242 响应 Body 参数

参数	参数类型	描述
snapshot_id	String	复制的快照ID。

请求示例

复制一个自动快照，复制的快照名为test1：

```
POST https://{Endpoint} /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/linked-copy
{
  "backup_name": "test1",
  "description": ""
}
```

响应示例

状态码： 200

```
{
  "snapshot_id":"52adf601-67f7-4639-bf71-061a61482869"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CopySnapshotSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CopySnapshotRequest request = new CopySnapshotRequest();
        LinkCopyReq body = new LinkCopyReq();
        body.withDescription("");
        body.withBackupName("test1");
        request.withBody(body);
        try {
            CopySnapshotResponse response = client.copySnapshot(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
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CopySnapshotRequest()
        request.body = LinkCopyReq(
            description="",
            backup_name="test1"
        )
        response = client.copy_snapshot(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CopySnapshotRequest{}
    descriptionLinkCopyReq:= ""
```

```

request.Body = &model.LinkCopyReq{
    Description: &descriptionLinkCopyReq,
    BackupName: "test1",
}
response, err := client.CopySnapshot(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	复制快照成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.15 查询集群快照列表

功能介绍

该接口用于查询集群快照列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/snapshots

表 5-243 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
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-244 Query 参数

参数	是否必选	参数类型	描述
sort_key	否	String	排序字段，排序字段需与排序规则一一对应。 可选字段： name、clusterName、startTime
sort_dir	否	String	排序规则： <ul style="list-style-type: none"> • ASC • DESC
offset	否	Integer	偏移量，从第几条开始查询。
limit	否	Integer	查询条数。

请求参数

无

响应参数

状态码： 200

表 5-245 响应 Body 参数

参数	参数类型	描述
snapshots	Array of Snapshots objects	快照对象列表。
count	Integer	快照对象列表总数
project_id	String	项目ID。
cluster_id	String	集群ID。

表 5-246 Snapshots

参数	参数类型	描述
id	String	快照ID。
name	String	快照名称。
description	String	快照描述。
started	String	快照创建的日期时间，格式为 ISO8601: YYYY-MM-DDThh:mm:ssZ。
updated	String	快照记录更新时间，格式为 ISO8601: YYYY-MM-DDThh:mm:ssZ。
size	Double	快照大小，单位 GB。
status	String	快照状态： <ul style="list-style-type: none"> ● CREATING: 创建中。 ● AVAILABLE: 可用。 ● UNAVAILABLE: 不可用，备份失败。 ● RESTORING: 恢复中。
type	String	快照创建类型。其中： <ul style="list-style-type: none"> ● MANUAL为手动快照。 ● AUTO为自动快照。
cluster_id	String	快照对应的集群ID。
cluster_name	String	快照对应的集群名称。
backup_key	String	备份产生的Key。
prior_backup_key	String	增量快照使用前一个快照的BackupKey。当取值为FULL时表示这是一个全量快照。
base_backup_key	String	对应全量快照的BackupKey。
backup_device	String	备份介质。
total_backup_size	Long	快照累计大小，表示从全量开始到当前增量的累计大小。单位KB
base_backup_name	String	对应全量快照名称。
support_inplace_restore	Boolean	是否支持恢复至原集群。
fine_grained_backup	Boolean	是否是细粒度备份。

参数	参数类型	描述
fine_grained_backup_detail	FineGrainedSnapshotDetail object	细粒度备份信息。
guest_agent_version	String	集群guestAgent版本号。
cluster_status	String	集群状态 <ul style="list-style-type: none"> • AVAILABLE: 可用 • UNAVAILABLE: 不可用 • FROZEN: 已冻结
bak_expected_start_time	String	预计快照开始时间。
bak_keep_day	Integer	备份保留天数。
bak_period	String	集群策略。
db_user	String	备份用户。
datastore	DataStore object	数据库版本
progress	String	备份进度。
backup_level	String	备份级别 <ul style="list-style-type: none"> • cluster • schema

表 5-247 FineGrainedSnapshotDetail

参数	参数类型	描述
database	String	备份数据库。
schema_list	Array of String	备份模式集合。
table_list	Array of String	备份表集合。

表 5-248 DataStore

参数	参数类型	描述
type	String	数据库类型。

参数	参数类型	描述
version	String	数据库版本。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/snapshots?sort_key=name&sort_dir=desc&offset=10&limit=5
```

响应示例

状态码： 200

查询快照列表成功。

```
{
  "snapshots": [
    {
      "datastore": {
        "type": "dws",
        "version": "8.1.3.200"
      },
      "description": "",
      "id": "02413539-6556-434c-a795-a44e9db7262d",
      "cluster_id": "1446b166-b73f-4096-b9bd-9a30afe3071f",
      "cluster_name": "ysj-1101-0917",
      "name": "kz-001",
      "size": 206400,
      "status": "COMPLETED",
      "updated": "2022-11-01T02:15:26",
      "type": "AUTO",
      "backup_key": "20221101_021217",
      "prior_backup_key": "FULL",
      "base_backup_key": "20221101_021217",
      "backup_device": "OBS",
      "total_backup_size": 206400,
      "base_backup_name": "kz-001",
      "support_inplace_restore": true,
      "fine_grained_backup": false,
      "fine_grained_backup_detail": {},
      "guest_agent_version": "8.2.0.1",
      "cluster_status": "AVAILABLE",
      "bak_expected_start_time": "1667295060000",
      "bak_keep_day": 3,
      "bak_period": "0 31 1/8 ? * 2,3,4,5,6,7",
      "db_user": "dbadmin"
    }
  ],
  "project_id": "4cf650fd46704908aa071b4df2453e1e",
  "cluster_id": "1446b166-b73f-4096-b9bd-9a30afe3071f",
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterSnapshotsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterSnapshotsRequest request = new ListClusterSnapshotsRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withSortKey("<sort_key>");
        request.withSortDir("<sort_dir>");
        try {
            ListClusterSnapshotsResponse response = client.listClusterSnapshots(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \
```

```
client = DwsClient.new_builder() \  
  .with_credentials(credentials) \  
  .with_region(DwsRegion.value_of("<YOUR REGION>")) \  
  .build()  
  
try:  
  request = ListClusterSnapshotsRequest()  
  request.limit = <limit>  
  request.offset = <offset>  
  request.sort_key = "<sort_key>"  
  request.sort_dir = "<sort_dir>"  
  response = client.list_cluster_snapshots(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"  
  dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
  region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
    dws.DwsClientBuilder().  
      WithRegion(region.ValueOf("<YOUR REGION>")).  
      WithCredential(auth).  
      Build())  
  
  request := &model.ListClusterSnapshotsRequest{}  
  limitRequest := int32(<limit>)  
  request.Limit = &limitRequest  
  offsetRequest := int32(<offset>)  
  request.Offset = &offsetRequest  
  sortKeyRequest := "<sort_key>"  
  request.SortKey = &sortKeyRequest  
  sortDirRequest := "<sort_dir>"  
  request.SortDir = &sortDirRequest  
  response, err := client.ListClusterSnapshots(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	查询快照列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.3.16 添加快照策略

功能介绍

该接口用于设置快照策略。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/snapshot-policies

表 5-249 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-250 请求 Body 参数

参数	是否必选	参数类型	描述
keep_day	否	Integer	保留天数。输入值必须在1-31之间。
backup_strategies	否	Array of BackupStrategyDetail objects	策略列表信息。当需要添加策略时该参数为必选。
device_name	否	String	备份设备。支持OBS和NFS。
server_port	否	String	端口。
backup_param	否	String	备份参数。
server_ips	否	Array of strings	备份介质服务IP。备份介质为NFS时该字段必填。表示NFS服务器地址。

表 5-251 BackupStrategyDetail

参数	是否必选	参数类型	描述
policy_id	否	String	策略ID。
policy_name	否	String	策略名称。添加备份策略时为必选字段。 <ul style="list-style-type: none"> 策略名称在4位到92位之间，必须以字母开头，不区分大小写，可以包含字母、数字、中划线或者下划线，不能包含其他特殊字符，并且名称唯一。
backup_strategy	否	String	执行策略。添加备份策略时为必选字段。符合Cron表达式格式。
backup_type	否	String	备份类型。添加备份策略时为必选字段。 <ul style="list-style-type: none"> increment 增量 full 全量
backup_level	否	String	备份级别。添加备份策略时为必选字段。 <ul style="list-style-type: none"> cluster

响应参数

无

请求示例

添加集群级别的全量快照策略，名为test，快照策略设置为周一到周日每天凌晨3点触发：

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/snapshot-policies
{
  "backup_strategies": [
    {
      "policy_name": "test",
      "backup_level": "cluster",
      "backup_type": "full",
      "backup_strategy": "0 0 3 ? * 1,2,3,4,5,6,7"
    }
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateSnapshotPolicySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateSnapshotPolicyRequest request = new CreateSnapshotPolicyRequest();
```

```
BackupPolicy body = new BackupPolicy();
request.withBody(body);
try {
    CreateSnapshotPolicyResponse response = client.createSnapshotPolicy(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateSnapshotPolicyRequest()
        request.body = BackupPolicy(
        )
        response = client.create_snapshot_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateSnapshotPolicyRequest{
    request.Body = &model.BackupPolicy{
    }
}
response, err := client.CreateSnapshotPolicy(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	添加快照策略成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.4 事件管理

5.4.1 查询事件列表

功能介绍

该接口用于查询事件列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/events

表 5-252 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 5-253 Query 参数

参数	是否必选	参数类型	描述
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

状态码： 200

表 5-254 响应 Body 参数

参数	参数类型	描述
events	Array of EventResponse objects	事件详情列表。
count	Integer	事件总数。

表 5-255 EventResponse

参数	参数类型	描述
category	String	事件类别。
description	String	事件描述。
event_id	String	事件ID。
name	String	事件定义名称。
display_name	String	事件显示名称。
name_space	String	所属服务。
severity	String	事件级别。
source_type	String	事件源类别。
occur_time	Long	时间。
project_id	String	项目ID。
source_id	String	事件源ID。
source_name	String	事件源名称。
status	Integer	状态。
subject	String	事件主题。
context	String	事件信息。

请求示例

https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/events

响应示例

查询事件列表成功。

```
{
  "events": [ {
    "category": "management",
    "description": "集群%s删除成功",
    "name": "deleteClusterSuccess",
    "severity": "normal",
    "status": 2,
    "subject": "DWS服务事件通知",
    "context": "集群test-ty-820-1006删除成功",
    "event_id": "f63ccf96-e3e0-474a-835a-fd1a779f68bd",
    "display_name": "集群删除成功",
    "name_space": "dws",
    "source_type": "cluster",
    "occur_time": 1664331248330,
    "project_id": "4cf650fd46704908aa071b4df2453e1e",
    "source_id": "9defa0ce-b11c-47b2-abbc-5cad09ced772",
    "source_name": "test-ty-820-1006"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListEventsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEventsRequest request = new ListEventsRequest();
        try {
            ListEventsResponse response = client.listEvents(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListEventsRequest()
    response = client.list_events(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEventsRequest{}
    response, err := client.ListEvents(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	查询事件列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.4.2 查询事件配置

功能介绍

该接口用于查询事件配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/event-specs

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 。

表 5-257 Query 参数

参数	是否必选	参数类型	描述
spec_name	否	String	事件配置名称。
category	否	String	事件类别。
severity	否	String	事件级别。
source_type	否	String	事件源类别。
tag	否	String	事件标签。

参数	是否必选	参数类型	描述
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

表 5-258 响应 Body 参数

参数	参数类型	描述
count	Integer	事件配置总数。
event_specs	Array of EventSpecResponse objects	事件配置列表。

表 5-259 EventSpecResponse

参数	参数类型	描述
id	String	事件配置ID。
name	String	事件配置定义名称。
display_name	String	事件配置显示名称。
description	String	事件配置描述。
subject	String	事件主题。
category	String	事件类别。
severity	String	事件级别。
source_type	String	事件源类型。
name_space	String	所属服务。

请求示例

https://{Endpoint}/v2/{project_id}/event-specs

响应示例

状态码： 200

集群查询事件配置成功。

```
{
  "event_specs" : [ {
    "id" : "fa6e1502-9d08-48c7-900c-26d3b5bd6078",
    "name" : "configureMRSExtDataSourcesSuccess",
    "description" : "配置集群%s的MRS外部数据源成功",
    "subject" : "DWS服务事件通知",
    "category" : "management",
    "severity" : "normal",
    "display_name" : "配置集群的MRS外部数据源成功",
    "source_type" : "cluster",
    "name_space" : "dws"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListEventSpecsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEventSpecsRequest request = new ListEventSpecsRequest();
        request.withSpecName("<spec_name>");
        request.withCategory("<category>");
        request.withSeverity("<severity>");
        request.withSourceType("<source_type>");
        request.withTag("<tag>");
        try {
            ListEventSpecsResponse response = client.listEventSpecs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListEventSpecsRequest()
        request.spec_name = "<spec_name>"
        request.category = "<category>"
        request.severity = "<severity>"
        request.source_type = "<source_type>"
        request.tag = "<tag>"
        response = client.list_event_specs(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListEventSpecsRequest{
    specNameRequest:= "<spec_name>"
    request.SpecName = &specNameRequest
    categoryRequest:= "<category>"
    request.Category = &categoryRequest
    severityRequest:= "<severity>"
    request.Severity = &severityRequest
    sourceTypeRequest:= "<source_type>"
    request.SourceType = &sourceTypeRequest
    tagRequest:= "<tag>"
    request.Tag = &tagRequest
}
response, err := client.ListEventSpecs(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	查询事件配置列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.4.3 查询订阅事件

功能介绍

该接口用于查询订阅的事件。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/event-sub

表 5-260 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 5-261 Query 参数

参数	是否必选	参数类型	描述
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

状态码： 200

表 5-262 响应 Body 参数

参数	参数类型	描述
count	Integer	事件订阅总数。
event_subscriptions	Array of EventSubscriptionResponse objects	事件订阅详情列表。

表 5-263 EventSubscriptionResponse

参数	参数类型	描述
id	String	订阅ID。
name	String	订阅名称。
source_type	String	事件源类型。
source_id	String	事件源ID。
category	String	事件类别。
severity	String	事件级别。
tag	String	事件标签。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息通知主题地址。
notification_target_name	String	消息通知主题名称。
notification_target_type	String	消息通知类型。
language	String	语言。
time_zone	String	时区。

请求示例

<https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/event-sub>

响应示例

状态码： 200

集群查询事件配置成功。

```
{
  "event_subscriptions": [ {
    "id": "4d62f33b-b9ee-41d3-b1bc-67e54b2239f9",
    "name": "00",
    "category": "",
    "severity": "",
    "tag": "",
    "enable": 1,
    "language": "zh-cn",
    "source_type": ""
  }
]
```

```
"source_id" : "",
"project_id" : "4cf650fd46704908aa071b4df2453e1e",
"name_space" : "DWS",
"notification_target" : "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:CGS",
"notification_target_name" : "CGS",
"notification_target_type" : "SMN",
"time_zone" : "GMT+08:00"
}],
"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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListEventSubsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEventSubsRequest request = new ListEventSubsRequest();
        try {
            ListEventSubsResponse response = client.listEventSubs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListEventSubsRequest()
        response = client.list_event_subs(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEventSubsRequest{}
    response, err := client.ListEventSubs(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	查询事件订阅列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.4.4 创建订阅事件

功能介绍

该接口用于添加订阅的事件。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/event-sub

表 5-264 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 5-265 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	事件订阅名称。
source_type	否	String	事件源类型支持： <ul style="list-style-type: none"> • cluster • backup • disaster-recovery
source_id	否	String	事件源ID。
category	否	String	事件类别支持： <ul style="list-style-type: none"> • management • monitor • security • system alarm 枚举值：多个枚举值使用 “, ” 分隔。
severity	否	String	事件级别支持normal, warning。
tag	否	String	事件标签。
enable	否	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启。 • 0为关闭。
notification_target	是	String	消息通知地址。
notification_target_name	是	String	消息主题名称。
notification_target_type	是	String	消息通知类型只支持SMN。
time_zone	否	String	时区。

响应参数

状态码： 200

表 5-266 响应 Body 参数

参数	参数类型	描述
id	String	订阅ID。
name	String	订阅名称。
source_type	String	事件源类型。
source_id	String	事件源ID。
category	String	事件类别。
severity	String	事件级别。
tag	String	事件标签。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息通知主题地址。
notification_target_name	String	消息通知主题名称。
notification_target_type	String	消息通知类型。
language	String	语言。
time_zone	String	时区。

请求示例

开启名为zrf-test-66的正常和警告事件订阅，并选择地址为“urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete”的SMN消息主题“dws-test-nodelete”：

https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/event-subs

```
{
  "severity": "normal,warning",
  "source_id": "",
  "source_type": "",
  "tag": "",
  "category": "",
  "enable": 1,
  "name": "zrf-test-66",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

响应示例

状态码： 200

```
{
  "id": "41eb162b-cd3b-4c66-88d0-0c2c17dfc2b",
  "name": "zrf-test-66",
  "category": "",
  "severity": "normal,warning",
  "tag": "",
  "enable": 1,
  "language": "zh-cn",
  "source_type": "",
  "source_id": "",
  "project_id": "4cf650fd46704908aa071b4df2453e1e",
  "name_space": "DWS",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateEventSubSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CreateEventSubRequest request = new CreateEventSubRequest();
        EventSubRequest body = new EventSubRequest();
        body.withTimeZone("GMT+08:00");
        body.withNotificationTargetType("SMN");
        body.withNotificationTargetName("dws-test-nodelete");
        body.withNotificationTarget("urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-
        nodelete");
        body.withEnable(1);
    }
}
```



```
body.withTag("");
body.withSeverity("normal,warning");
body.withCategory("");
body.withSourceId("");
body.withSourceType("");
body.withName("zrf-test-66");
request.withBody(body);
try {
    CreateEventSubResponse response = client.createEventSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateEventSubRequest()
        request.body = EventSubRequest(
            time_zone="GMT+08:00",
            notification_target_type="SMN",
            notification_target_name="dws-test-nodelete",
            notification_target="urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
            enable=1,
            tag="",
            severity="normal,warning",
            category="",
            source_id="",
            source_type="",
            name="zrf-test-66"
        )
        response = client.create_event_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateEventSubRequest{}
    timeZoneEventSubRequest := "GMT+08:00"
    enableEventSubRequest := int32(1)
    tagEventSubRequest := ""
    severityEventSubRequest := "normal,warning"
    categoryEventSubRequest := ""
    sourceIdEventSubRequest := ""
    sourceTypeEventSubRequest := ""
    request.Body = &model.EventSubRequest{
        TimeZone: &timeZoneEventSubRequest,
        NotificationTargetType: "SMN",
        NotificationTargetName: "dws-test-nodelete",
        NotificationTarget: "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
        Enable: &enableEventSubRequest,
        Tag: &tagEventSubRequest,
        Severity: &severityEventSubRequest,
        Category: &categoryEventSubRequest,
        SourceId: &sourceIdEventSubRequest,
        SourceType: &sourceTypeEventSubRequest,
        Name: "zrf-test-66",
    }
    response, err := client.CreateEventSub(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	创建事件订阅成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.4.5 更新订阅事件

功能介绍

该接口用于更新订阅事件。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/event-sub/{event_sub_id}

表 5-267 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 。
event_sub_id	是	String	事件订阅ID。

请求参数

表 5-268 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	事件订阅名称。

参数	是否必选	参数类型	描述
source_type	否	String	事件源类型支持 <ul style="list-style-type: none"> • cluster • backup • disaster-recovery
source_id	否	String	事件源ID。
category	否	String	事件类别支持 <ul style="list-style-type: none"> • management • monitor • security • system alarm 枚举值：多个枚举值使用“，”分隔。
severity	否	String	事件级别支持normal和warning。
tag	否	String	事件标签。
enable	否	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
notification_target	是	String	消息通知地址。
notification_target_name	是	String	消息主题名称。
notification_target_type	是	String	消息通知类型只支持SMN。

响应参数

状态码： 200

表 5-269 响应 Body 参数

参数	参数类型	描述
id	String	订阅ID。
name	String	订阅名称。
source_type	String	事件源类型。
source_id	String	事件源ID。

参数	参数类型	描述
category	String	事件类别。
severity	String	事件级别。
tag	String	事件标签。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息通知主题地址。
notification_target_name	String	消息通知主题名称。
notification_target_type	String	消息通知类型。
language	String	语言。
time_zone	String	时区。

请求示例

更新zrf-test-66事件订阅，事件类别修改为正常和警告，SMN消息主题dws-test-nodelete地址更新为“urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete”：

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/event-subs/41eb162b-cd3b-4c66-88d0-0c2c17fdcf2b
```

```
{
  "severity": "normal,warning",
  "source_id": "",
  "source_type": "",
  "tag": "",
  "category": "",
  "enable": 1,
  "name": "zrf-test-66",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN"
}
```

响应示例

状态码： 200

```
{
  "id": "41eb162b-cd3b-4c66-88d0-0c2c17fdcf2b",
  "name": "zrf-test-66",
  "category": "",

```

```
"severity" : "normal,warning",
"tag" : "",
"enable" : 1,
"language" : "zh-cn",
"source_type" : "",
"source_id" : "",
"project_id" : "4cf650fd46704908aa071b4df2453e1e",
"name_space" : "DWS",
"notification_target" : "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
"notification_target_name" : "dws-test-nodelete",
"notification_target_type" : "SMN",
"time_zone" : "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateEventSubSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        UpdateEventSubRequest request = new UpdateEventSubRequest();
        EventSubUpdateRequest body = new EventSubUpdateRequest();
        body.withNotificationTargetType("SMN");
        body.withNotificationTargetName("dws-test-nodelete");
        body.withNotificationTarget("urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-
nodelete");
        body.withEnable(1);
        body.withTag("");
        body.withSeverity("normal,warning");
        body.withCategory("");
        body.withSourceId("");
        body.withSourceType("");
        body.withName("zrf-test-66");
        request.withBody(body);
        try {
            UpdateEventSubResponse response = client.updateEventSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateEventSubRequest()
        request.body = EventSubUpdateRequest(
            notification_target_type="SMN",
            notification_target_name="dws-test-nodelete",
            notification_target="urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
            enable=1,
            tag="",
            severity="normal,warning",
            category="",
            source_id="",
            source_type="",
            name="zrf-test-66"
        )
        response = client.update_event_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEventSubRequest{
        enableEventSubUpdateRequest:= int32(1)
        tagEventSubUpdateRequest:= ""
        severityEventSubUpdateRequest:= "normal,warning"
        categoryEventSubUpdateRequest:= ""
        sourceIdEventSubUpdateRequest:= ""
        sourceTypeEventSubUpdateRequest:= ""
        request.Body = &model.EventSubUpdateRequest{
            NotificationTargetType: "SMN",
            NotificationTargetName: "dws-test-nodelete",
            NotificationTarget: "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
            Enable: &enableEventSubUpdateRequest,
            Tag: &tagEventSubUpdateRequest,
            Severity: &severityEventSubUpdateRequest,
            Category: &categoryEventSubUpdateRequest,
            SourceId: &sourceIdEventSubUpdateRequest,
            SourceType: &sourceTypeEventSubUpdateRequest,
            Name: "zrf-test-66",
        }
    }
    response, err := client.UpdateEventSub(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	更新事件订阅成功。
400	请求错误。
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
501	服务不可用。

5.4.6 删除订阅事件

功能介绍

该接口用于删除订阅的事件。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/event-sub/{event_sub_id}
```

表 5-270 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 。
event_sub_id	是	String	事件订阅ID。

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
id	String	订阅ID。
name	String	订阅名称。
source_type	String	事件源类型。

参数	参数类型	描述
source_id	String	事件源ID。
category	String	事件类别。
severity	String	事件级别。
tag	String	事件标签。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息通知主题地址。
notification_target_name	String	消息通知主题名称。
notification_target_type	String	消息通知类型。
language	String	语言。
time_zone	String	时区。

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/event-subs/41eb162b-cd3b-4c66-88d0-0c2c17fdcf2b
```

响应示例

状态码： 200

```
{
  "id": "41eb162b-cd3b-4c66-88d0-0c2c17fdcf2b",
  "name": "zrf-test-66",
  "category": "",
  "severity": "normal,warning",
  "tag": "",
  "enable": 1,
  "language": "zh-cn",
  "source_type": "",
  "source_id": "",
  "project_id": "4cf650fd46704908aa071b4df2453e1e",
  "name_space": "DWS",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteEventSubSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        DeleteEventSubRequest request = new DeleteEventSubRequest();
        try {
            DeleteEventSubResponse response = client.deleteEventSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = DeleteEventSubRequest()
    response = client.delete_event_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteEventSubRequest{}
    response, err := client.DeleteEventSub(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	删除事件订阅成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.5 数据源

5.5.1 查询数据源

功能介绍

该接口用于查询数据源。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/ext-data-sources

表 5-272 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-273 响应 Body 参数

参数	参数类型	描述
data_sources	Array of ExtDataSource objects	数据源列表。
project_id	String	项目ID。
cluster_id	String	集群ID。
type	String	数据源类型。其中包含： <ul style="list-style-type: none"> • OBS数据源 • MRS数据源
count	Integer	总数。

表 5-274 ExtDataSource

参数	参数类型	描述
id	String	数据源配置ID。
name	String	数据源名称。
type	String	数据源类型。
connect_info	String	连接的数据库。
user_name	String	MRS数据源用户名，OBS数据源对应委托名称。
version	String	外部数据源版本。
configure_status	String	配置状态。
status	String	状态。
data_source_id	String	外部数据源ID。
created	String	数据源创建时间。
updated	String	数据源更新时间。
data_source_updated	String	数据源更新时间。
extend_properties	Map<String,String>	扩展信息。
description	String	描述。
fail_reason	String	失败原因。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/ext-data-sources?type=OBS
```

响应示例

```
{
  "data_sources": [
    {
      "id": "8010dbc0-c1d2-48d6-87eb-3229f9195fed",
      "name": "tes54",
      "type": "OBS",
      "connect_info": "gaussdb",
      "user_name": "DWSAccessOBS",
      "configure_status": "300",
      "created": "2022-10-30T07:38:33",
      "extend_properties": {
      },
      "description": "xxx",
      "fail_reason": ""
    }
  ],
  "project_id": "0536cdee2200d5912f7cc00b877980f1",
  "cluster_id": "a07cb2f7-b17e-4d95-923b-a33d0c884d37",
  "type": "OBS",
  "count": 5
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListDataSourceSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDataSourceRequest request = new ListDataSourceRequest();
```

```
try {
    ListDataSourceResponse response = client.listDataSource(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDataSourceRequest()
        response = client.list_data_source(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListDataSourceRequest{}
response, err := client.ListDataSource(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	查询数据源成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.5.2 创建数据源

功能介绍

该接口用于创建一个数据源。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/ext-data-sources

表 5-275 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-276 请求 Body 参数

参数	是否必选	参数类型	描述
data_source_id	否	String	外部数据源ID。当数据源为MRS时为必选字段。获取方法，请参见 查询数据源 。
type	是	String	数据源类型，其中包含： <ul style="list-style-type: none"> • OBS • MRS
data_source_name	是	String	数据源名称。仅可包含大小写字母，数字，下划线，1到64个字符。
user_name	是	String	用户名。仅可包含大小写字母，数字，下划线，3到20个字符。数据源类型为OBS时，传对应OBS委托名称。
user_pwd	否	String	密码。当数据源为MRS时为必选字段
description	否	String	描述。除!<>'=&等特殊字符之外的字符。长度256个字符以内。
reboot	否	Boolean	是否重启。
connect_info	否	String	连接的数据库。当数据源为OBS时为必选字段。

响应参数

表 5-277 响应 Body 参数

参数	参数类型	描述
id	String	数据源配置ID。
job_id	String	创建数据源的任务ID。

请求示例

创建名为test154的OBS数据源，用户名为“DWSAccessOBS”，连接gaussdb数据库：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/ext-data-sources
{
  "type": "OBS",
  "data_source_name": "test154",
  "user_name": "DWSAccessOBS",
  "connect_info": "gaussdb",
  "description": "xxx"
}
```

响应示例

状态码： 200

```
{
  "id" : "d303f19c-fbd1-4716-ae04-07d1ff358f22",
  "job_id" : "2c9081c88418ed420184197c8bc80001"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateDataSourceSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
CreateDataSourceRequest request = new CreateDataSourceRequest();
ExtDataSourceReq body = new ExtDataSourceReq();
body.withConnectInfo("gaussdb");
body.withDescription("xxx");
body.withUserName("DWSAccessOBS");
body.withDataSourceName("test154");
body.withType("OBS");
request.withBody(body);
try {
    CreateDataSourceResponse response = client.createDataSource(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateDataSourceRequest()
        request.body = ExtDataSourceReq(
            connect_info="gaussdb",
            description="xxx",
            user_name="DWSAccessOBS",
            data_source_name="test154",
            type="OBS"
        )
```

```
response = client.create_data_source(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateDataSourceRequest{
        connectInfoExtDataSourceReq:= "gaussdb"
        descriptionExtDataSourceReq:= "xxx"
        request.Body = &model.ExtDataSourceReq{
            ConnectInfo: &connectInfoExtDataSourceReq,
            Description: &descriptionExtDataSourceReq,
            UserName: "DWSAccessOBS",
            DataSourceName: "test154",
            Type: "OBS",
        }
    }
    response, err := client.CreateDataSource(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	创建数据源成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.5.3 更新数据源

功能介绍

该接口用于更新一个数据源。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1.0/{project_id}/clusters/{cluster_id}/ext-data-sources/{ext_data_source_id}

表 5-278 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
ext_data_source_id	是	String	数据源配置ID。获取方法，请参见 查询数据源 。

请求参数

表 5-279 请求 Body 参数

参数	是否必选	参数类型	描述
reconfigure	是	ReConfigure object	配置信息。

表 5-280 ReConfigure

参数	是否必选	参数类型	描述
reboot	否	Boolean	是否重启。
agency	否	String	委托名称。获取方法，请参见 查询数据源 。

响应参数

表 5-281 响应参数

参数	参数类型	描述
job_id	String	更新数据源的任务ID。

请求示例

修改ID为“d303f19c-fbd1-4716-ae04-07d1ff358f22”数据源的委托名称为“DWSAccessOBS”：

```
PUT https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/ext-data-sources/d303f19c-fbd1-4716-ae04-07d1ff358f22
{
  "reconfigure":{
    "reboot": false,
    "agency": "DWSAccessOBS"
  }
}
```

响应示例

状态码： 200

```
{
  "job_id": "2c9081c88418ed420184197c8bc80001"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateDataSourceSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        UpdateDataSourceRequest request = new UpdateDataSourceRequest();
        ReconfigureExtDataSourceActionReq body = new ReconfigureExtDataSourceActionReq();
        request.withBody(body);
        try {
            UpdateDataSourceResponse response = client.updateDataSource(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = UpdateDataSourceRequest()
    request.body = ReconfigureExtDataSourceActionReq(
    )
    response = client.update_data_source(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDataSourceRequest{}
    request.Body = &model.ReconfigureExtDataSourceActionReq{
    }
    response, err := client.UpdateDataSource(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	更新数据源请求成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.5.4 删除数据源

功能介绍

该接口用于删除数据源。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v1.0/{project_id}/clusters/{cluster_id}/ext-data-sources/{ext_data_source_id}
```

表 5-282 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
ext_data_source_id	是	String	数据源配置ID。获取方法，请参见 查询数据源 。

请求参数

无

响应参数

表 5-283 响应参数

参数	参数类型	描述
job_id	String	更新数据源的任务ID。

请求示例

```
DELETE https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-  
b4f3-0ad4e5e3ba90/ext-data-sources/d303f19c-fbd1-  
4716-ae04-07d1ff358f22
```

响应示例

状态码： 200

```
{  
  "job_id" : "2c9081c88418ed420184197c8bc80001"  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class DeleteDataSourceSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("cn-north-4"))  
            .build();  
        DeleteDataSourceRequest request = new DeleteDataSourceRequest();  
        try {  
            DeleteDataSourceResponse response = client.deleteDataSource(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteDataSourceRequest()
        response = client.delete_data_source(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.DeleteDataSourceRequest{}
response, err := client.DeleteDataSource(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	删除数据源成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.6 审计日志

5.6.1 查询审计日志记录

功能介绍

该接口用于查询审计日志记录。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/clusters/{cluster_id}/audit-log-records

表 5-284 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-285 响应 Body 参数

参数	参数类型	描述
records	Array of AuditDumpRecord objects	审计日志列表。
cluster_id	String	集群ID。
count	Integer	总数。

表 5-286 AuditDumpRecord

参数	参数类型	描述
cluster_id	String	集群ID。
executor_time	String	执行时间。
begin_time	String	开始时间。
end_time	String	结束时间。
bucket_name	String	桶名。
location_prefix	String	OBS路径。
result	String	结果状态。

参数	参数类型	描述
failed_reason	String	失败原因。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/audit-log-records
```

响应示例

```
{
  "records": [
    {
      "result": "RUNNING",
      "cluster_id": "a07cb2f7-b17e-4d95-923b-a33d0c884d37",
      "executor_time": "2022-10-31T09:11:31",
      "begin_time": "2022-10-31T09:09:55",
      "end_time": "2022-10-31T09:19:55",
      "bucket_name": "dws-demo",
      "location_prefix": "test"
    }
  ],
  "count": 1,
  "cluster_id": "a07cb2f7-b17e-4d95-923b-a33d0c884d37"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAuditLogSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        ListAuditLogRequest request = new ListAuditLogRequest();
        try {
            ListAuditLogResponse response = client.listAuditLog(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAuditLogRequest()
        response = client.list_audit_log(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAuditLogRequest{}
response, err := client.ListAuditLog(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	查询审计日志记录成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7 资源管理

5.7.1 设置资源管理开关

功能介绍

该接口用于设置资源管理。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload

表 5-287 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-288 请求 Body 参数

参数	是否必选	参数类型	描述
workload_status	否	WorkloadStatus object	资源管理状态。

表 5-289 WorkloadStatus

参数	是否必选	参数类型	描述
workload_switch	是	String	开关。取值：on/off。
max_concurrency_num	否	String	最大并发数。-1~2147483647 范围之间的整数，其中-1/0表示不限制。

响应参数

表 5-290 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	返回状态。
workload_res_str	String	状态描述。

请求示例

开启资源管理，并设置最大并发数为5：

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-
b4f3-0ad4e5e3ba90/workload
{
  "workload_status":{
    "workload_switch":"on",
    "max_concurrency_num":"5"
  }
}
```

响应示例

状态码： 200

设置资源管理成功

```
{
  "workload_res_code": 0,
  "workload_res_str": "Success to modify the resource manager config."
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateClusterWorkloadSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateClusterWorkloadRequest request = new CreateClusterWorkloadRequest();
        WorkloadStatusReq body = new WorkloadStatusReq();
        request.withBody(body);
        try {
            CreateClusterWorkloadResponse response = client.createClusterWorkload(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateClusterWorkloadRequest()
        request.body = WorkloadStatusReq(
        )
        response = client.create_cluster_workload(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateClusterWorkloadRequest{
    request.Body = &model.WorkloadStatusReq{
    }
}
response, err := client.CreateClusterWorkload(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	设置资源管理成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.2 查询资源管理状态

功能介绍

该接口用于查询资源管理状态。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload

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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群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
workload_status	WorkloadStatus object	资源管理状态。
workload_res_code	Integer	返回状态。
workload_res_str	String	状态描述。

表 5-293 WorkloadStatus

参数	参数类型	描述
workload_switch	String	资源管理开关。 有效值包括： <ul style="list-style-type: none"> on: 已开启。 off: 已关闭。
max_concurrency_num	String	最大并发数。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload
```

响应示例

```
{
  "workload_res_code": 0,
  "workload_res_str": "Success get workload manager status",
  "workload_status": {
    "workload_switch": "on",
    "max_concurrency_num": 60
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterWorkloadSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterWorkloadRequest request = new ListClusterWorkloadRequest();
        try {
            ListClusterWorkloadResponse response = client.listClusterWorkload(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterWorkloadRequest()
        response = client.list_cluster_workload(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```



```
Build()  
  
request := &model.ListClusterWorkloadRequest{}  
response, err := client.ListClusterWorkload(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	查询资源管理状态成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.3 查询资源统计

功能介绍

该接口用于查询资源统计。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/resource-statistics

表 5-294 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 5-295 Query 参数

参数	是否必选	参数类型	描述
namespace	否	String	命名空间。

请求参数

无

响应参数

状态码： 200

表 5-296 响应 Body 参数

参数	参数类型	描述
cluster_statistics	StatusStatistics object	集群资源统计信息。
node_statistics	StatusStatistics object	节点资源统计信息。

表 5-297 StatusStatistics

参数	参数类型	描述
active	Long	活跃资源。
total	Long	总资源。

请求示例

GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/resource-statistics

响应示例

状态码： 200

查询资源统计成功

```
{
  "cluster_statistics": {
    "active": 2,
    "total": 2
  },
  "node_statistics": {
    "active": 6,
    "total": 36
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowResourceStatisticsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowResourceStatisticsRequest request = new ShowResourceStatisticsRequest();
        request.withNamespace("<namespace>");
        try {
            ShowResourceStatisticsResponse response = client.showResourceStatistics(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ShowResourceStatisticsRequest()
    request.namespace = "<namespace>"
    response = client.show_resource_statistics(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowResourceStatisticsRequest{}
    namespaceRequest := "<namespace>"
    request.Namespace = &namespaceRequest
    response, err := client.ShowResourceStatistics(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	查询资源统计成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.7.4 添加资源池

功能介绍

该接口用于添加资源池。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/workload/queues

表 5-298 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-299 请求 Body 参数

参数	是否必选	参数类型	描述
workload_queue	是	WorkloadQueue object	资源池对象。

表 5-300 WorkloadQueue

参数	是否必选	参数类型	描述
workload_queue_name	是	String	资源池名称。以小写字母开头，只包含小写字母、数字、下划线(_)，长度3-28个字符。
logical_cluster_name	否	String	逻辑集群名称。非逻辑集群模式下该字段不填，逻辑集群模式下需指定逻辑集群名称。
workload_resource_item_list	是	Array of WorkloadResource objects	资源配置队列。

表 5-301 WorkloadResource

参数	是否必选	参数类型	描述
resource_name	是	String	<p>资源名称。</p> <ul style="list-style-type: none"> memory: 内存资源 (%) tablespace: 存储资源 (MB) activestatements: 查询并发 cpu_limit: 专属限额 cpu_share: 共享配额 <p>说明 添加资源池时，所有的资源类型都要填写。其中cpu_limit 仅限在8.1.3以上集群支持。</p>

参数	是否必选	参数类型	描述
resource_value	是	Integer	资源属性值。 <ul style="list-style-type: none"> memory 取值0 ~ 100，其中0表示不管控。 tablespace 取值-1 ~ 2147483647，其中-1表示不限制，注：该值取值应参考实际存储资源。 activestatements 取值-1 ~ 2147483647 其中-1/0表示不限制。 cpu_limit 取值1~99。cpu_limit为0则不约束。 cpu_share 取值1~99。cpu_share不填写默认为20。 cpu_limit与cpu_share支持同时设置时，以cpu_limit专属限额优先。

响应参数

表 5-302 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	返回状态。
workload_res_str	String	状态描述。

请求示例

添加资源池test11，选择共享配额设为12%，内存资源设为0（不限制），存储资源设为-1（不限制），查询并发设为10：

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues
{
  "workload_queue": {
    "workload_queue_name": "test11",
    "workload_resource_item_list": [
      {
        "resource_name": "memory",
        "resource_value": "0"
      },
      {
        "resource_name": "tablespace",
        "resource_value": "-1"
      }
    ]
  }
}
```

```

    "resource_name": "activestatements",
    "resource_value": "10"
  },
  {
    "resource_name": "cpu_limit",
    "resource_value": 0
  },
  {
    "resource_name": "cpu_share",
    "resource_value": 12
  }
],
"logical_cluster_name": ""
}
}

```

响应示例

状态码： 200

添加资源池状态成功。

```

{
  "workload_res_code" : 0,
  "workload_res_str" : "Success to create the resource pool."
}

```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class AddWorkloadQueueSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
    }
}

```



```
AddWorkloadQueueRequest request = new AddWorkloadQueueRequest();
WorkloadQueueReq body = new WorkloadQueueReq();
List<WorkloadResource> listWorkloadQueueWorkloadResourceItemList = new ArrayList<>();
listWorkloadQueueWorkloadResourceItemList.add(
    new WorkloadResource()
        .withResourceName("memory")
        .withResourceValue(0)
);
listWorkloadQueueWorkloadResourceItemList.add(
    new WorkloadResource()
        .withResourceName("tablespace")
        .withResourceValue(-1)
);
listWorkloadQueueWorkloadResourceItemList.add(
    new WorkloadResource()
        .withResourceName("activestatements")
        .withResourceValue(10)
);
listWorkloadQueueWorkloadResourceItemList.add(
    new WorkloadResource()
        .withResourceName("cpu_limit")
        .withResourceValue(0)
);
listWorkloadQueueWorkloadResourceItemList.add(
    new WorkloadResource()
        .withResourceName("cpu_share")
        .withResourceValue(12)
);
WorkloadQueue workloadQueuebody = new WorkloadQueue();
workloadQueuebody.withWorkloadQueueName("test11")
    .withLogicalClusterName("")
    .withWorkloadResourceItemList(listWorkloadQueueWorkloadResourceItemList);
body.withWorkloadQueue(workloadQueuebody);
request.withBody(body);
try {
    AddWorkloadQueueResponse response = client.addWorkloadQueue(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = AddWorkloadQueueRequest()
    listWorkloadResourceItemWorkloadQueue = [
        WorkloadResource(
            resource_name="memory",
            resource_value=0
        ),
        WorkloadResource(
            resource_name="tablespace",
            resource_value=-1
        ),
        WorkloadResource(
            resource_name="activestatements",
            resource_value=10
        ),
        WorkloadResource(
            resource_name="cpu_limit",
            resource_value=0
        ),
        WorkloadResource(
            resource_name="cpu_share",
            resource_value=12
        )
    ]
    workloadQueuebody = WorkloadQueue(
        workload_queue_name="test11",
        logical_cluster_name="",
        workload_resource_item_list=listWorkloadResourceItemWorkloadQueue
    )
    request.body = WorkloadQueueReq(
        workload_queue=workloadQueuebody
    )
    response = client.add_workload_queue(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.AddWorkloadQueueRequest{}
var listWorkloadResourceItemWorkloadQueue = []model.WorkloadResource{
    {
        ResourceName: "memory",
        ResourceValue: int32(0),
    },
    {
        ResourceName: "tablespace",
        ResourceValue: int32(-1),
    },
    {
        ResourceName: "activestatements",
        ResourceValue: int32(10),
    },
    {
        ResourceName: "cpu_limit",
        ResourceValue: int32(0),
    },
    {
        ResourceName: "cpu_share",
        ResourceValue: int32(12),
    },
}
logicalClusterNameWorkloadQueue:= ""
workloadQueuebody := &model.WorkloadQueue{
    WorkloadQueueName: "test11",
    LogicalClusterName: &logicalClusterNameWorkloadQueue,
    WorkloadResourceItemWorkloadQueue: listWorkloadResourceItemWorkloadQueue,
}
request.Body = &model.WorkloadQueueReq{
    WorkloadQueue: workloadQueuebody,
}
response, err := client.AddWorkloadQueue(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	添加资源池成功。
400	请求错误。
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.5 删除资源池

功能介绍

该接口用于删除资源池。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/clusters/{cluster_id}/workload/queues

表 5-303 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-304 Query 参数

参数	是否必选	参数类型	描述
logical_cluster_name	否	String	逻辑集群名称。逻辑集群模式下该字段必填。
workload_queue_name	是	String	资源池名称。

请求参数

无

响应参数

表 5-305 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
DELETE https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/workload/queues?logical_cluster_name=lc_test&workload_queue_name=queue1
```

响应示例

状态码： 200

删除资源池成功。

```
{
  "workload_res_code": 0,
  "workload_res_str": "删除资源池成功。"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteWorkloadQueueSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(DwsRegion.valueOf("cn-north-4"))
        .build();
DeleteWorkloadQueueRequest request = new DeleteWorkloadQueueRequest();
request.withLogicalClusterName("<logical_cluster_name>");
request.withWorkloadQueueName("<workload_queue_name>");
try {
    DeleteWorkloadQueueResponse response = client.deleteWorkloadQueue(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteWorkloadQueueRequest()
        request.logical_cluster_name = "<logical_cluster_name>"
        request.workload_queue_name = "<workload_queue_name>"
        response = client.delete_workload_queue(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
```

```

region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteWorkloadQueueRequest{}
    logicalClusterNameRequest := "<logical_cluster_name>"
    request.LogicalClusterName = &logicalClusterNameRequest
    request.WorkloadQueueName = "<workload_queue_name>"
    response, err := client.DeleteWorkloadQueue(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	删除资源池成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.6 查询资源池列表

功能介绍

该接口用于查询资源池列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload/queues

表 5-306 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-307 Query 参数

参数	是否必选	参数类型	描述
logical_cluster_name	否	String	逻辑集群名称。逻辑集群模式下该字段必填。

请求参数

无

响应参数

状态码： 200

表 5-308 响应 Body 参数

参数	参数类型	描述
workload_queue_name_list	Array of strings	资源池名称。
workload_res_code	Integer	响应编码，其中“0”表示成功。

参数	参数类型	描述
workload_res_str	String	响应描述。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues
```

响应示例

```
{
  "workload_res_code": 0,
  "workload_res_str": "Success get workload manager queue name list",
  "workload_queue_name_list": [
    "test1"
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListWorkloadQueueSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkloadQueueRequest request = new ListWorkloadQueueRequest();
        try {
            ListWorkloadQueueResponse response = client.listWorkloadQueue(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListWorkloadQueueRequest()
        response = client.list_workload_queue(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListWorkloadQueueRequest{}
response, err := client.ListWorkloadQueue(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	查询资源池列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.7 获得资源池详细信息

功能介绍

该接口用于获得资源池详细信息。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v2/{project_id}/clusters/{cluster_id}/workload/queues/{queue_name}
```

表 5-309 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
queue_name	是	String	资源池名称。

表 5-310 Query 参数

参数	是否必选	参数类型	描述
logical_cluster_name	否	String	逻辑集群名称。非逻辑集群模式下该字段不填，逻辑集群模式下需指定逻辑集群名称。

请求参数

无

响应参数

状态码： 200

表 5-311 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	结果状态码。
workload_res_str	String	结果描述。
workload_queue	WorkloadQueueItem object	资源池详情。

表 5-312 WorkloadQueueItem

参数	参数类型	描述
queue_name	String	资源池名称。
logical_cluster_name	String	逻辑集群名称。

参数	参数类型	描述
short_query_optimize	String	资源池短查询加速开关。
short_query_concurrency_num	Integer	资源池短查询并发数。
resource_item_list	Array of WorkloadResourceItem objects	资源配置队列。

表 5-313 WorkloadResourceItem

参数	参数类型	描述
resource_name	String	资源名称。
resource_value	Integer	资源属性值。
value_unit	String	资源属性单位。
resource_description	String	资源附加描述。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues/resource1
```

响应示例

状态码： 200

获得资源池详细信息成功。

```
{
  "workload_res_code": 0,
  "workload_res_str": "获取资源池信息成功。",
  "workload_queue": {
    "queue_name": "resource1",
    "logical_cluster_name": "",
    "short_query_optimize": "t",
    "short_query_concurrency_num": -1,
    "resource_item_list": [ {
      "resource_name": "cpu",
      "resource_value": 1,
      "value_unit": null,
      "resource_description": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowWorkloadQueueSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowWorkloadQueueRequest request = new ShowWorkloadQueueRequest();
        try {
            ShowWorkloadQueueResponse response = client.showWorkloadQueue(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ShowWorkloadQueueRequest()
    response = client.show_workload_queue(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowWorkloadQueueRequest{}
    response, err := client.ShowWorkloadQueue(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	获得资源池详细信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.8 更新资源池配置信息

功能介绍

该接口用于更新资源池资源配置信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/workload/queues/{queue_name}/resources

表 5-314 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
queue_name	是	String	资源池名称。

请求参数

表 5-315 请求 Body 参数

参数	是否必选	参数类型	描述
workload_queue	是	WorkloadQueueInfo object	资源池信息。

表 5-316 WorkloadQueueInfo

参数	是否必选	参数类型	描述
workload_queue_name	是	String	资源池名称。
logical_cluster_name	否	String	逻辑集群名称。
resource_item_list	是	Array of WorkloadResourceItem objects	资源配置队列。

表 5-317 WorkloadResourceItem

参数	是否必选	参数类型	描述
resource_name	是	String	资源名称。
resource_value	是	Integer	资源属性值。
value_unit	否	String	资源属性单位。
resource_description	否	String	资源附加描述。

响应参数

状态码： 200

表 5-318 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues/{queue_name}/resources

```
{
  "workload_queue": {
    "workload_queue_name": "test11",
    "resource_item_list": [ {
      "resource_name": "memory",
      "resource_value": "0"
    }, {
      "resource_name": "tablespace",
      "resource_value": "-1"
    }, {
      "resource_name": "activestatements",
      "resource_value": "10"
    }, {
      "resource_name": "cpu_limit",
      "resource_value": 0
    }, {
      "resource_name": "cpu_share",
      "resource_value": 12
    }
  ],
  "logical_cluster_name": ""
}
```

响应示例

状态码： 200

更新资源池资源配置信息成功。

```
{
  "workload_res_code": 0,
  "workload_res_str": "修改资源池成功。"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class UpdateQueueResourcesSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateQueueResourcesRequest request = new UpdateQueueResourcesRequest();
        WorkloadQueueRequest body = new WorkloadQueueRequest();
        List<WorkloadResourceItem> listWorkloadQueueResourceItemList = new ArrayList<>();
        listWorkloadQueueResourceItemList.add(
            new WorkloadResourceItem()
                .withResourceName("memory")
                .withResourceValue(0)
        );
        listWorkloadQueueResourceItemList.add(
            new WorkloadResourceItem()
                .withResourceName("tablespace")
                .withResourceValue(-1)
        );
        listWorkloadQueueResourceItemList.add(
            new WorkloadResourceItem()
                .withResourceName("activestatements")
                .withResourceValue(10)
        );
        listWorkloadQueueResourceItemList.add(
            new WorkloadResourceItem()
                .withResourceName("cpu_limit")
                .withResourceValue(0)
        );
        listWorkloadQueueResourceItemList.add(
            new WorkloadResourceItem()
                .withResourceName("cpu_share")
                .withResourceValue(12)
        );
        WorkloadQueueInfo workloadQueuebody = new WorkloadQueueInfo();
        workloadQueuebody.withWorkloadQueueName("test11")
            .withLogicalClusterName("")
            .withResourceItemList(listWorkloadQueueResourceItemList);
        body.withWorkloadQueue(workloadQueuebody);
        request.withBody(body);
        try {
            UpdateQueueResourcesResponse response = client.updateQueueResources(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateQueueResourcesRequest()
        listResourceItemWorkloadQueue = [
            WorkloadResourceItem(
                resource_name="memory",
                resource_value=0
            ),
            WorkloadResourceItem(
                resource_name="tablespace",
                resource_value=-1
            ),
            WorkloadResourceItem(
                resource_name="activestatements",
                resource_value=10
            ),
            WorkloadResourceItem(
                resource_name="cpu_limit",
                resource_value=0
            ),
            WorkloadResourceItem(
                resource_name="cpu_share",
                resource_value=12
            )
        ]
        workloadQueuebody = WorkloadQueueInfo(
            workload_queue_name="test11",
            logical_cluster_name="",
            resource_item_list=listResourceItemWorkloadQueue
        )
        request.body = WorkloadQueueRequest(
            workload_queue=workloadQueuebody
        )
        response = client.update_queue_resources(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateQueueResourcesRequest{}
    var listResourceItemWorkloadQueue = []model.WorkloadResourceItem{
        {
            ResourceName: "memory",
            ResourceValue: int32(0),
        },
        {
            ResourceName: "tablespace",
            ResourceValue: int32(-1),
        },
        {
            ResourceName: "activestatements",
            ResourceValue: int32(10),
        },
        {
            ResourceName: "cpu_limit",
            ResourceValue: int32(0),
        },
        {
            ResourceName: "cpu_share",
            ResourceValue: int32(12),
        },
    }
    logicalClusterNameWorkloadQueue := ""
    workloadQueuebody := &model.WorkloadQueueInfo{
        WorkloadQueueName: "test11",
        LogicalClusterNameWorkloadQueue: &logicalClusterNameWorkloadQueue,
        ResourceItemWorkloadQueue: listResourceItemWorkloadQueue,
    }
    request.Body = &model.WorkloadQueueRequest{
        WorkloadQueue: workloadQueuebody,
    }
    response, err := client.UpdateQueueResources(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	更新资源池资源配置信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.9 资源池关联用户

功能介绍

该接口用于添加资源池关联用户。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/queues/{queue_name}/users/batch-create

表 5-319 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

参数	是否必选	参数类型	描述
queue_name	是	String	资源池名称。

请求参数

表 5-320 请求 Body 参数

参数	是否必选	参数类型	描述
queue_name	是	String	资源池名称。
user_list	是	Array of user_list objects	资源池用户列表。

表 5-321 user_list

参数	是否必选	参数类型	描述
user_name	否	String	用户名。

响应参数

状态码： 200

表 5-322 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues/resource1/users/batch-create

```
{
  "queue_name": "test11",
  "user_list": [ {
    "user_name": "user_batch"
  } ]
}
```

响应示例

状态码： 200

添加资源池的绑定用户成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "Success to associate users."
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class AddQueueUserListSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        AddQueueUserListRequest request = new AddQueueUserListRequest();
        WorkloadQueueUserReq body = new WorkloadQueueUserReq();
        List<WorkloadQueueUserReqUserList> listbodyUserList = new ArrayList<>();
        listbodyUserList.add(
            new WorkloadQueueUserReqUserList()
                .withUserName("user_batch")
        );
        body.withUserList(listbodyUserList);
        body.withQueueName("test11");
        request.withBody(body);
        try {
            AddQueueUserListResponse response = client.addQueueUserList(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddQueueUserListRequest()
        listUserListbody = [
            WorkloadQueueUserReqUserList(
                user_name="user_batch"
            )
        ]
        request.body = WorkloadQueueUserReq(
            user_list=listUserListbody,
            queue_name="test11"
        )
        response = client.add_queue_user_list(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddQueueUserListRequest{
    userNameUserList:= "user_batch"
    var listUserListbody = []model.WorkloadQueueUserReqUserList{
        {
            UserName: &userNameUserList,
        },
    }
    request.Body = &model.WorkloadQueueUserReq{
        UserList: listUserListbody,
        QueueName: "test11",
    }
    response, err := client.AddQueueUserList(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	添加资源池的绑定用户成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.10 获得资源池绑定用户列表

功能介绍

该接口用于获得资源池绑定用户列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload/queues/{queue_name}/users

表 5-323 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
queue_name	是	String	资源池名称。

表 5-324 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

请求参数

无

响应参数

状态码： 200

表 5-325 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	结果状态码。

参数	参数类型	描述
workload_res_str	String	结果描述。
queue_name	String	资源池名称。
user_list	Array of WorkloadQueueUser objects	资源池用户列表。
count	Integer	总数量。

表 5-326 WorkloadQueueUser

参数	参数类型	描述
user_name	String	用户名。
occupy_resource_list	Array of OccupyResource objects	执行计划阶段。
exec_result	Integer	执行结果。
exec_log	String	执行日志。

表 5-327 OccupyResource

参数	参数类型	描述
resource_name	String	资源项名称。
resource_value	Integer	资源属性数值。
value_unit	String	资源属性单位。
resource_description	String	资源附加描述。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues/resource1/users
```

响应示例

状态码： 200

获得资源池的绑定用户列表成功。

```
{
  "queue_name": "rs1",
  "user_list": [ {
    "user_name": "user_batch",
    "occupy_resource_list": [ {
      "resource_name": "memory_used",
      "resource_value": 0,
      "value_unit": "M",
      "resource_description": ""
    } ]
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListWorkloadQueueUsersSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkloadQueueUsersRequest request = new ListWorkloadQueueUsersRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListWorkloadQueueUsersResponse response = client.listWorkloadQueueUsers(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListWorkloadQueueUsersRequest()
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_workload_queue_users(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
```

```

dws.DwsClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build()

request := &model.ListWorkloadQueueUsersRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
response, err := client.ListWorkloadQueueUsers(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	获得资源池的绑定用户列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.11 资源池解绑用户

功能介绍

该接口用于资源池解绑用户。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/queues/{queue_name}/users/batch-delete

表 5-328 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
queue_name	是	String	资源池名称。

请求参数

表 5-329 请求 Body 参数

参数	是否必选	参数类型	描述
queue_name	是	String	资源池名称。
user_list	是	Array of user_list objects	资源池用户列表。

表 5-330 user_list

参数	是否必选	参数类型	描述
user_name	否	String	用户名。

响应参数

状态码： 200

表 5-331 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/queues/resource1/users/batch-delete
```



```
{
  "queue_name": "test11",
  "user_list": [ {
    "user_name": "user_batch"
  } ]
}
```

响应示例

状态码： 200

删除资源池的绑定用户成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "取消关联用户成功。"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class DeleteQueueUserListSolution {

    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);

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

        DeleteQueueUserListRequest request = new DeleteQueueUserListRequest();
        WorkloadQueueUserReq body = new WorkloadQueueUserReq();
        List<WorkloadQueueUserReqUserList> listbodyUserList = new ArrayList<>();
        listbodyUserList.add(
            new WorkloadQueueUserReqUserList()
                .withUserName("user_batch")
        );
        body.withUserList(listbodyUserList);
        body.withQueueName("test11");
        request.withBody(body);
    }
}
```

```
try {
    DeleteQueueUserListResponse response = client.deleteQueueUserList(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteQueueUserListRequest()
        listUserListbody = [
            WorkloadQueueUserReqUserList(
                user_name="user_batch"
            )
        ]
        request.body = WorkloadQueueUserReq(
            user_list=listUserListbody,
            queue_name="test11"
        )
        response = client.delete_queue_user_list(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteQueueUserListRequest{}
    userNameUserList:= "user_batch"
    var listUserListbody = []model.WorkloadQueueUserReqUserList{
        {
            UserName: &userNameUserList,
        },
    }
    request.Body = &model.WorkloadQueueUserReq{
        UserList: listUserListbody,
        QueueName: "test11",
    }
    response, err := client.DeleteQueueUserList(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	删除资源池的绑定用户成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.7.12 添加资源管理计划

功能介绍

该接口用于添加资源管理计划。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/plans

表 5-332 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-333 请求 Body 参数

参数	是否必选	参数类型	描述
plan_name	是	String	资源管理计划名称。以小写字母开头只能包含字母、数字、下划线(_)，长度3-28位。
logical_cluster_name	否	String	逻辑集群名称。

响应参数

表 5-334 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	返回状态。
workload_res_str	String	状态描述。

请求示例

添加资源管理计划test2:

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans
{
  "plan_name":"test2"
}
```

响应示例

状态码： 200

添加资源管理计划成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "success added workload plan"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateWorkloadPlanSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
CreateWorkloadPlanRequest request = new CreateWorkloadPlanRequest();
WorkloadPlanReq body = new WorkloadPlanReq();
body.withPlanName("test2");
request.withBody(body);
try {
    CreateWorkloadPlanResponse response = client.createWorkloadPlan(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateWorkloadPlanRequest()
        request.body = WorkloadPlanReq(
            plan_name="test2"
        )
        response = client.create_workload_plan(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateWorkloadPlanRequest{}
    request.Body = &model.WorkloadPlanReq{
        PlanName: "test2",
    }
    response, err := client.CreateWorkloadPlan(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	添加资源管理计划成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.7.13 查询资源管理计划列表

功能介绍

该接口用于查询集群中所有资源管理计划。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload/plans

表 5-335 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-336 Query 参数

参数	是否必选	参数类型	描述
logical_cluster_name	否	String	逻辑集群名称。非逻辑集群模式下该字段不填，逻辑集群模式下需指定逻辑集群名称。
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

请求参数

无

响应参数

状态码： 200

表 5-337 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	结果状态码。
workload_res_str	String	结果描述。
plan_list	Array of WorkloadPlanInfo objects	资源池名称。
count	Integer	总数量。

表 5-338 WorkloadPlanInfo

参数	参数类型	描述
status	Integer	计划状态。
project_id	String	项目ID。
cluster_id	String	集群ID。
plan_id	String	计划ID。
plan_name	String	计划名称。
current_stage	String	当前计划阶段。
logical_cluster_name	String	逻辑集群名称。
stage_list	Array of PlanStage objects	计划阶段列表。

表 5-339 PlanStage

参数	参数类型	描述
month	String	计划月份。
day	String	计划日期。
plan_id	String	计划ID。
stage_id	String	计划阶段ID。
stage_name	String	计划阶段名称。
start_time	String	计划开始时间。

参数	参数类型	描述
end_time	String	计划结束时间。
next_valid_time	String	下次校验时间。
queue_list	Array of QueueResourceItem objects	资源队列列表。

表 5-340 QueueResourceItem

参数	参数类型	描述
queue_name	String	资源池名称。
queue_resources	Array of WorkloadResourceItem objects	资源配置队列。

表 5-341 WorkloadResourceItem

参数	参数类型	描述
resource_name	String	资源名称。
resource_value	Integer	资源属性值。
value_unit	String	资源属性单位。
resource_description	String	资源附加描述。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans
```

响应示例

状态码： 200

查询资源管理计划列表成功。

```
{
  "plan_list": [ {
    "status": 0,
```

```
"cluster_id" : "862201cd-f822-4dbc-ae3b-c7b9d080eea6",
"project_id" : "0536cdee2200d5912f7cc00b877980f1",
"plan_id" : "dde42d0c-38f7-4592-841d-5a16c5512408",
"plan_name" : "plan1",
"current_stage" : null,
"logical_cluster_name" : null,
"stage_list" : null
}],
"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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListWorkloadPlansSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListWorkloadPlansRequest request = new ListWorkloadPlansRequest();
        request.withLogicalClusterName("<logical_cluster_name>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListWorkloadPlansResponse response = client.listWorkloadPlans(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListWorkloadPlansRequest()
        request.logical_cluster_name = "<logical_cluster_name>"
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_workload_plans(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ListWorkloadPlansRequest{}
logicalClusterNameRequest:= "<logical_cluster_name>"
request.LogicalClusterName = &logicalClusterNameRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
response, err := client.ListWorkloadPlans(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	查询资源管理计划列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.14 启动资源管理计划

功能介绍

该接口用于启动资源管理计划。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/start

表 5-342 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

请求参数

无

响应参数

状态码： 200

表 5-343 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a/start
```

响应示例

状态码： 200

启动资源管理计划成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "success started workload plan"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class StartWorkloadPlanSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        StartWorkloadPlanRequest request = new StartWorkloadPlanRequest();
        try {
            StartWorkloadPlanResponse response = client.startWorkloadPlan(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(DwsRegion.value_of("cn-north-4")) \
.build()

try:
    request = StartWorkloadPlanRequest()
    response = client.start_workload_plan(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.StartWorkloadPlanRequest{}
    response, err := client.StartWorkloadPlan(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	启动资源管理计划成功。

状态码	描述
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.15 查看计划执行日志

功能介绍

该接口用于查看计划执行日志。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/logs

表 5-344 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

表 5-345 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

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
workload_res_code	Integer	结果状态码。
workload_res_str	String	结果描述。
plan_logs	Array of PlanLog objects	资源池名称。
count	Integer	总数量。

表 5-347 PlanLog

参数	参数类型	描述
exec_time	String	执行时间。
stage_info	String	执行计划阶段。
exec_result	Integer	执行结果。
exec_log	String	执行日志。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a/logs
```

响应示例

状态码： 200

查询资源管理计划列表成功。

```
{
  "plan_logs": [ {
    "exec_time": "2023-08-23 13:28:50",
    "stage_info": "stage1",
    "exec_result": 0,
    "exec_log": "2023-08-23 13:28:47.661892+00:00 UTC |INFO| start change stage."
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListPlanExecLogsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPlanExecLogsRequest request = new ListPlanExecLogsRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListPlanExecLogsResponse response = client.listPlanExecLogs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListPlanExecLogsRequest()
    request.limit = <limit>
    request.offset = <offset>
    response = client.list_plan_exec_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlanExecLogsRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListPlanExecLogs(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	查询资源管理计划列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.16 查询某个资源管理计划详细信息

功能介绍

该接口用于查询某个资源管理计划详细信息。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}
```

表 5-348 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。获取方法，请参见 查询资源管理计划列表 。

请求参数

无

响应参数

状态码： 200

表 5-349 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	结果状态码。
workload_res_str	String	结果描述。
workload_plan	WorkloadPlanInfo object	队列详情。

表 5-350 WorkloadPlanInfo

参数	参数类型	描述
status	Integer	计划状态。 有效值包括： <ul style="list-style-type: none"> • 1: 计划运行中。 • 0: 计划未启动。
project_id	String	项目ID。
cluster_id	String	集群ID。
plan_id	String	计划ID。
plan_name	String	计划名称。
current_stage	String	当前计划阶段。
logical_cluster_name	String	逻辑集群名称。
stage_list	Array of PlanStage objects	计划阶段列表。

表 5-351 PlanStage

参数	参数类型	描述
month	String	计划月份。
day	String	计划日期。
plan_id	String	计划ID。
stage_id	String	计划阶段ID。
stage_name	String	计划阶段名称。
start_time	String	计划开始时间。
end_time	String	计划结束时间。
next_valid_time	String	下次校验时间。
queue_list	Array of QueueResourceItem objects	资源队列列表。

表 5-352 QueueResourceItem

参数	参数类型	描述
queue_name	String	资源池名称。
queue_resources	Array of WorkloadResourceItem objects	资源配置队列。

表 5-353 WorkloadResourceItem

参数	参数类型	描述
resource_name	String	资源名称。
resource_value	Integer	资源属性值。
value_unit	String	资源属性单位。
resource_description	String	资源附加描述

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a
```

响应示例

状态码： 200

查询某个资源管理计划详细信息成功。

```
{
  "workload_plan": {
    "status": 0,
    "cluster_id": "862201cd-f822-4dbc-ae3b-c7b9d080eea6",
    "project_id": "0536cdee2200d5912f7cc00b877980f1",
    "plan_id": "dde42d0c-38f7-4592-841d-5a16c5512408",
    "plan_name": "plan1",
    "current_stage": "",
    "logical_cluster_name": "",
    "stage_list": [ {
      "month": 4,
      "day": 4,
      "plan_id": "dde42d0c-38f7-4592-841d-5a16c5512408",
      "stage_id": "00bc8bc5-c885-42e5-b205-80547d3f94c4",
      "stage_name": "stage2",
      "start_time": "00:00:00",
      "end_time": "00:00:00",
      "next_valid_time": "2024-04-04 00:00:00",
      "queue_list": [ {
        "queue_name": "resouce1",
        "queue_resources": [ {
          "resource_name": "cpu",
          "resource_value": 1,
          "value_unit": "%",
          "resource_description": "Workload Queue Memory Percent"
        }
      ]
    }
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowWorkloadPlanSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ShowWorkloadPlanRequest request = new ShowWorkloadPlanRequest();
try {
    ShowWorkloadPlanResponse response = client.showWorkloadPlan(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowWorkloadPlanRequest()
        response = client.show_workload_plan(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowWorkloadPlanRequest{}
    response, err := client.ShowWorkloadPlan(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	查询某个资源管理计划详细信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.17 停止资源管理计划

功能介绍

该接口用于停止资源管理计划。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/stop
```

表 5-354 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

请求参数

无

响应参数

状态码： 200

表 5-355 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a/stop
```

响应示例

状态码： 200

停止资源管理计划成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "success stopped workload plan"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class StopWorkloadPlanSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        StopWorkloadPlanRequest request = new StopWorkloadPlanRequest();
        try {
            StopWorkloadPlanResponse response = client.stopWorkloadPlan(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = StopWorkloadPlanRequest()
        response = client.stop_workload_plan(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.StopWorkloadPlanRequest{}
    response, err := client.StopWorkloadPlan(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	停止资源管理计划成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.18 删除资源管理计划

功能介绍

该接口用于删除资源管理计划。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}
```

表 5-356 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

请求参数

无

响应参数

状态码： 200

表 5-357 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
DELETE https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a
```

响应示例

状态码： 200

删除资源管理计划成功。

```
{
  "workload_res_code" : 0,
  "workload_res_str" : "success deleted workload plan"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteWorkloadPlanSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
DeleteWorkloadPlanRequest request = new DeleteWorkloadPlanRequest();
try {
    DeleteWorkloadPlanResponse response = client.deleteWorkloadPlan(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteWorkloadPlanRequest()
        response = client.delete_workload_plan(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteWorkloadPlanRequest{}
    response, err := client.DeleteWorkloadPlan(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	删除资源管理计划成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.19 添加资源管理计划阶段

功能介绍

该接口用于添加资源管理计划阶段。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/stages

表 5-358 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

请求参数

表 5-359 请求 Body 参数

参数	是否必选	参数类型	描述
workload_plan_stage	否	workload_plan_stage object	资源管理计划阶段详情。

表 5-360 workload_plan_stage

参数	是否必选	参数类型	描述
day	否	String	日期。
month	否	String	月份。
stage_name	否	String	计划阶段。
start_time	否	String	开始时间。
end_time	否	String	结束时间。

参数	是否必选	参数类型	描述
queue_list	否	Array of QueueResourceItem objects	资源队列。

表 5-361 QueueResourceItem

参数	是否必选	参数类型	描述
queue_name	是	String	资源池名称。
queue_resources	是	Array of WorkloadResourceItem objects	资源配置队列。

表 5-362 WorkloadResourceItem

参数	是否必选	参数类型	描述
resource_name	是	String	资源名称。
resource_value	是	Integer	资源属性值。
value_unit	否	String	资源属性单位。
resource_description	否	String	资源附加描述。

响应参数

状态码： 200

表 5-363 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0591129f-4970-44e7-b213-f7bce797de87/stages

```
{
  "workload_plan_stage": {
    "day": "1,2",
    "month": "6",
    "stage_name": "stage1",
    "start_time": "00:00:00",
    "end_time": "00:00:00",
    "queue_list": [ {
      "queue_name": "pool1",
      "queue_resources": [ {
        "resource_name": "cpu",
        "resource_value": 20,
        "value_unit": null,
        "resource_description": null
      }, {
        "resource_name": "cpu_limit",
        "resource_value": 0,
        "value_unit": null,
        "resource_description": null
      }, {
        "resource_name": "memory",
        "resource_value": 0,
        "value_unit": null,
        "resource_description": null
      }, {
        "resource_name": "concurrency",
        "resource_value": 10,
        "value_unit": null,
        "resource_description": null
      }, {
        "resource_name": "shortQueryConcurrencyNum",
        "resource_value": -1,
        "value_unit": null,
        "resource_description": null
      }
    ]
  }
}
```

响应示例

状态码： 200

添加资源管理计划阶段成功。

```
{
  "workload_res_code": 0,
  "workload_res_str": ""
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class AddWorkloadPlanStageSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        AddWorkloadPlanStageRequest request = new AddWorkloadPlanStageRequest();
        WorkloadPlanStageReq body = new WorkloadPlanStageReq();
        List<WorkloadResourceItem> listQueueListQueueResources = new ArrayList<>();
        listQueueListQueueResources.add(
            new WorkloadResourceItem()
                .withResourceName("cpu")
                .withResourceValue(20)
        );
        listQueueListQueueResources.add(
            new WorkloadResourceItem()
                .withResourceName("cpu_limit")
                .withResourceValue(0)
        );
        listQueueListQueueResources.add(
            new WorkloadResourceItem()
                .withResourceName("memory")
                .withResourceValue(0)
        );
        listQueueListQueueResources.add(
            new WorkloadResourceItem()
                .withResourceName("concurrency")
                .withResourceValue(10)
        );
        listQueueListQueueResources.add(
            new WorkloadResourceItem()
                .withResourceName("shortQueryConcurrencyNum")
                .withResourceValue(-1)
        );
        List<QueueResourceItem> listWorkloadPlanStageQueueList = new ArrayList<>();
        listWorkloadPlanStageQueueList.add(
            new QueueResourceItem()
                .withQueueName("pool1")
                .withQueueResources(listQueueListQueueResources)
        );
        WorkloadPlanStageReqWorkloadPlanStage workloadPlanStagebody = new
        WorkloadPlanStageReqWorkloadPlanStage();
        workloadPlanStagebody.withDay("1,2")
            .withMonth("6")
            .withStageName("stage1")
            .withStartTime("00:00:00")
            .withEndTime("00:00:00")
            .withQueueList(listWorkloadPlanStageQueueList);
    }
}
```

```
body.withWorkloadPlanStage(workloadPlanStagebody);
request.withBody(body);
try {
    AddWorkloadPlanStageResponse response = client.addWorkloadPlanStage(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddWorkloadPlanStageRequest()
        listQueueResourcesQueueList = [
            WorkloadResourceItem(
                resource_name="cpu",
                resource_value=20
            ),
            WorkloadResourceItem(
                resource_name="cpu_limit",
                resource_value=0
            ),
            WorkloadResourceItem(
                resource_name="memory",
                resource_value=0
            ),
            WorkloadResourceItem(
                resource_name="concurrency",
                resource_value=10
            ),
            WorkloadResourceItem(
                resource_name="shortQueryConcurrencyNum",
                resource_value=-1
            )
        ]
        listQueueListWorkloadPlanStage = [
```

```

        QueueResourceItem(
            queue_name="pool1",
            queue_resources=listQueueResourcesQueueList
        )
    ]
    workloadPlanStagebody = WorkloadPlanStageReqWorkloadPlanStage(
        day="1,2",
        month="6",
        stage_name="stage1",
        start_time="00:00:00",
        end_time="00:00:00",
        queue_list=listQueueListWorkloadPlanStage
    )
    request.body = WorkloadPlanStageReq(
        workload_plan_stage=workloadPlanStagebody
    )
    response = client.add_workload_plan_stage(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"
    dws "github.com/ HuaweiCloud/ HuaweiCloud-sdk-go-v3/services/dws/v2"
    "github.com/ HuaweiCloud/ HuaweiCloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/ HuaweiCloud/ HuaweiCloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddWorkloadPlanStageRequest{}
    var listQueueResourcesQueueList = []model.WorkloadResourceItem{
        {
            ResourceName: "cpu",
            ResourceValue: int32(20),
        },
        {
            ResourceName: "cpu_limit",
            ResourceValue: int32(0),
        },
        {
            ResourceName: "memory",
            ResourceValue: int32(0),
        }
    }
}

```

```

    },
    {
      ResourceName: "concurrency",
      ResourceValue: int32(10),
    },
    {
      ResourceName: "shortQueryConcurrencyNum",
      ResourceValue: int32(-1),
    },
  },
}
var listQueueListWorkloadPlanStage = []model.QueueResourceItem{
  {
    QueueName: "pool1",
    QueueResources: listQueueResourcesQueueList,
  },
}
dayWorkloadPlanStage:= "1,2"
monthWorkloadPlanStage:= "6"
stageNameWorkloadPlanStage:= "stage1"
startTimeWorkloadPlanStage:= "00:00:00"
endTimeWorkloadPlanStage:= "00:00:00"
workloadPlanStagebody := &model.WorkloadPlanStageReqWorkloadPlanStage{
  Day: &dayWorkloadPlanStage,
  Month: &monthWorkloadPlanStage,
  StageName: &stageNameWorkloadPlanStage,
  StartTime: &startTimeWorkloadPlanStage,
  EndTime: &endTimeWorkloadPlanStage,
  QueueList: &listQueueListWorkloadPlanStage,
}
request.Body = &model.WorkloadPlanStageReq{
  WorkloadPlanStage: workloadPlanStagebody,
}
response, err := client.AddWorkloadPlanStage(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	添加资源管理计划阶段成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.20 查询资源管理计划阶段详细信息

功能介绍

该接口用于查询资源管理计划阶段详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/stages/{stage_id}

表 5-364 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。
stage_id	是	String	计划阶段ID。

请求参数

无

响应参数

状态码： 200

表 5-365 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	结果状态码。
workload_res_str	String	结果描述。
workload_plan_stage	PlanStage object	计划阶段详情。

表 5-366 PlanStage

参数	参数类型	描述
month	String	计划月份。
day	String	计划日期。
plan_id	String	计划ID。
stage_id	String	计划阶段ID。
stage_name	String	计划阶段名称。
start_time	String	计划开始时间。
end_time	String	计划结束时间
next_valid_time	String	下次校验时间
queue_list	Array of QueueResourceItem objects	资源队列列表

表 5-367 QueueResourceItem

参数	参数类型	描述
queue_name	String	资源池名称。
queue_resources	Array of WorkloadResourceItem objects	资源配置队列。

表 5-368 WorkloadResourceItem

参数	参数类型	描述
resource_name	String	资源名称。
resource_value	Integer	资源属性值。
value_unit	String	资源属性单位。
resource_description	String	资源附加描述

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a/stages/00bc8bc5-c885-42e5-b205-80547d3f94c4
```

响应示例

状态码： 200

查询资源管理计划阶段详细信息成功。

```
{
  "workload_plan_stage" : {
    "month" : 4,
    "day" : 4,
    "plan_id" : "d4e42d0c-38f7-4592-841d-5a16c5512408",
    "stage_id" : "00bc8bc5-c885-42e5-b205-80547d3f94c4",
    "stage_name" : "stage2",
    "start_time" : "00:00:00",
    "end_time" : "00:00:00",
    "next_valid_time" : "2024-04-04 00:00:00",
    "queue_list" : [ {
      "queue_name" : "resouce1",
      "queue_resources" : [ {
        "resource_name" : "cpu",
        "resource_value" : 1,
        "value_unit" : null,
        "resource_description" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowWorkloadPlanStageSolution {

    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);
```

```
DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ShowWorkloadPlanStageRequest request = new ShowWorkloadPlanStageRequest();
try {
    ShowWorkloadPlanStageResponse response = client.showWorkloadPlanStage(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowWorkloadPlanStageRequest()
        response = client.show_workload_plan_stage(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowWorkloadPlanStageRequest{}
    response, err := client.ShowWorkloadPlanStage(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	查询资源管理计划阶段详细信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.21 删除资源管理计划阶段

功能介绍

该接口用于删除资源管理计划阶段。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/stages/{stage_id}
```

表 5-369 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。
stage_id	是	String	计划阶段ID。

请求参数

无

响应参数

状态码： 200

表 5-370 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
DELETE https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0c2145ad-4d76-4abe-bd1b-cdbe9128478a/stages/00bc8bc5-c885-42e5-b205-80547d3f94c4
```

响应示例

状态码： 200

删除资源管理计划阶段成功。

```
{  
  "workload_res_code" : 0,  
}
```

```
"workload_res_str" : "success deleted workload plan stage"  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class DeleteWorkloadPlanStageSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        DeleteWorkloadPlanStageRequest request = new DeleteWorkloadPlanStageRequest();  
        try {  
            DeleteWorkloadPlanStageResponse response = client.deleteWorkloadPlanStage(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteWorkloadPlanStageRequest()
        response = client.delete_workload_plan_stage(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteWorkloadPlanStageRequest{}
    response, err := client.DeleteWorkloadPlanStage(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	删除资源管理计划阶段成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.22 切换资源管理计划阶段

功能介绍

该接口用于切换资源管理计划阶段。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/workload/plans/{plan_id}/stage-switch

表 5-371 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
plan_id	是	String	计划ID。

请求参数

表 5-372 请求 Body 参数

参数	是否必选	参数类型	描述
stage_id	是	String	计划阶段ID。

响应参数

状态码： 200

表 5-373 响应 Body 参数

参数	参数类型	描述
workload_res_code	Integer	响应编码。
workload_res_str	String	响应信息。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/workload/plans/0591129f-4970-44e7-b213-f7bce797de87/stage-switch
{
  "stage_id": "0c2145ad-4d76-4abe-bd1b-cdbe9128478a"
}
```

响应示例

状态码： 200

切换资源管理计划阶段成功。

```
{
  "workload_res_code": 0,
  "workload_res_str": ""
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class SwitchPlanStageSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        SwitchPlanStageRequest request = new SwitchPlanStageRequest();
        WorkloadPlanStageIdReq body = new WorkloadPlanStageIdReq();
        body.withStageId("0c2145ad-4d76-4abe-bd1b-cdbe9128478a");
        request.withBody(body);
        try {
            SwitchPlanStageResponse response = client.switchPlanStage(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

```
try:
    request = SwitchPlanStageRequest()
    request.body = WorkloadPlanStageIdReq(
        stage_id="0c2145ad-4d76-4abe-bd1b-cdbe9128478a"
    )
    response = client.switch_plan_stage(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchPlanStageRequest{}
    request.Body = &model.WorkloadPlanStageIdReq{
        StageId: "0c2145ad-4d76-4abe-bd1b-cdbe9128478a",
    }
    response, err := client.SwitchPlanStage(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	切换资源管理计划阶段成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.23 查询集群模式空间信息

功能介绍

该接口用于查询集群模式空间信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/databases/{database_name}/schemas

表 5-374 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
database_name	是	String	数据库名称。

表 5-375 Query 参数

参数	是否必选	参数类型	描述
sort_key	否	String	排序字段。
sort_dir	否	String	排序规则。

参数	是否必选	参数类型	描述
keywords	否	String	查询关键词。
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

请求参数

无

响应参数

状态码： 200

表 5-376 响应 Body 参数

参数	参数类型	描述
schemas	Array of SchemaInfo objects	集群模式空间信息列表。
count	Integer	总数量。

表 5-377 SchemaInfo

参数	参数类型	描述
schema_name	String	Schema名称。
database_name	String	数据库名称。
total_value	Integer	集群Schema使用空间总值。
perm_space	Integer	Schema空间阈值。
skew_percent	Double	倾斜率。
min_value	Integer	最小值。
max_value	Integer	最大值。
min_dn	String	最小dn节点。
max_dn	String	最大cn节点。
dn_num	Integer	dn节点数量。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/databases/gaussdb/schemas
```

响应示例

状态码： 200

查询集群模式空间信息列表成功。

```
{
  "schemas": [ {
    "schema_name": "gs_logical_cluster",
    "database_name": "gaussdb",
    "total_value": 0,
    "perm_space": -1,
    "skew_percent": 0.0,
    "min_value": 0,
    "max_value": 0,
    "min_dn": "dn_6001_6002",
    "max_dn": "",
    "dn_num": 3
  } ],
  "count": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListSchemasSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListSchemasRequest request = new ListSchemasRequest();
        request.withSortKey("<sort_key>");
        request.withSortDir("<sort_dir>");
    }
}
```

```
request.withKeywords("<keywords>");
request.withLimit(<limit>);
request.withOffset(<offset>);
try {
    ListSchemasResponse response = client.listSchemas(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListSchemasRequest()
        request.sort_key = "<sort_key>"
        request.sort_dir = "<sort_dir>"
        request.keywords = "<keywords>"
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_schemas(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
```



```

region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSchemasRequest{}
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    sortDirRequest:= "<sort_dir>"
    request.SortDir = &sortDirRequest
    keywordsRequest:= "<keywords>"
    request.Keywords = &keywordsRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListSchemas(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	查询集群模式空间信息列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.7.24 更新模式空间限额

功能介绍

该接口用于更新模式空间限额。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/databases/{database_name}/schemas

表 5-378 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
database_name	是	String	数据库名称。

请求参数

表 5-379 请求 Body 参数

参数	是否必选	参数类型	描述
schema_name	是	String	模式空间名称。
perm_space	是	String	Schema空间阈值。

响应参数

状态码： 200

表 5-380 响应 Body 参数

参数	参数类型	描述
ret_code	Integer	响应编码。

请求示例

```
PUT https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/databases/gaussdb/schemas
{
  "schema_name": "gs_logical_cluster",
  "perm_space": 10240
}
```

响应示例

状态码： 200

更新模式空间限额成功。

```
{
  "ret_code": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateSchemasSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSchemasRequest request = new UpdateSchemasRequest();
        WorkloadSchemaReq body = new WorkloadSchemaReq();
        body.withPermSpace("10240");
        body.withSchemaName("gs_logical_cluster");
        request.withBody(body);
        try {
            UpdateSchemasResponse response = client.updateSchemas(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateSchemasRequest()
        request.body = WorkloadSchemaReq(
            perm_space="10240",
            schema_name="gs_logical_cluster"
        )
        response = client.update_schemas(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateSchemasRequest{}
request.Body = &model.WorkloadSchemaReq{
    PermSpace: "10240",
    SchemaName: "gs_logical_cluster",
}
response, err := client.UpdateSchemas(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	更新模式空间限额成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8 告警管理

5.8.1 查询告警配置

功能介绍

该接口用于查询告警配置。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/alarm-configs

表 5-381 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 5-382 Query 参数

参数	是否必选	参数类型	描述
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

状态码： 200

表 5-383 响应 Body 参数

参数	参数类型	描述
count	Integer	告警配置总数。
alarm_configs	Array of AlarmConfig Response objects	告警配置列表。

表 5-384 AlarmConfigResponse

参数	参数类型	描述
id	String	告警配置ID。

参数	参数类型	描述
alarm_id	String	告警ID。
alarm_name	String	告警名称。
name_space	String	所属服务。
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none"> urgent: 紧急 important: 重要 minor: 次要 prompt: 提示
is_user_visible	String	用户是否可见。
is_converge	String	是否覆盖。
converge_time	Integer	覆盖时间。
is_maintain_visible	String	运维是否可见。

请求示例

```
https://{Endpoint}/v2/{project_id}/alarm-configs
```

响应示例

状态码： 200

```
{
  "alarm_configs": [ {
    "id": "fd02e440-b4e2-4d2c-8d98-4d80224cf848",
    "alarm_id": "DWS_2000000021_1",
    "alarm_name": "文件句柄数使用率超阈值",
    "name_space": "dws",
    "alarm_level": "urgent",
    "is_user_visible": "1",
    "is_converge": "0",
    "converge_time": 0,
    "is_maintain_visible": "0"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAlarmConfigsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAlarmConfigsRequest request = new ListAlarmConfigsRequest();
        try {
            ListAlarmConfigsResponse response = client.listAlarmConfigs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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



```
try:
    request = ListAlarmConfigsRequest()
    response = client.list_alarm_configs(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAlarmConfigsRequest{}
    response, err := client.ListAlarmConfigs(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	查询告警配置成功。
400	请求错误。

状态码	描述
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.2 查询告警详情列表

功能介绍

该接口用于查询告警详情列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/alarms

表 5-385 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 5-386 Query 参数

参数	是否必选	参数类型	描述
time_zone	是	String	时区。
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

状态码： 200

表 5-387 响应 Body 参数

参数	参数类型	描述
count	Integer	告警详情总数。
alarm_details	Array of AlarmDetailResponse objects	告警列表。

表 5-388 AlarmDetailResponse

参数	参数类型	描述
alarm_id	String	告警定义ID。
alarm_name	String	告警名称。
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none"> urgent: 紧急 important: 重要 minor: 次要 prompt: 提示
alarm_source	String	告警服务。
alarm_message	String	告警消息。
alarm_location	String	告警定位信息。
resource_id	String	告警源ID。
resource_id_name	String	告警源名称。
alarm_generate_date	String	告警日期。
alarm_status	String	告警状态。

请求示例

https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarms?time_zone=GMT

响应示例

状态码： 200

```
{
  "alarm_details": [ {
    "alarm_id": "DWS_01010",
    "alarm_name": "集群状态异常",
    "alarm_level": "1",
    "alarm_source": "DWS",
    "alarm_message": "CloudService=DWS, resourceId: 5e76e8e2-d0cf-4b64-9d9a-aadbb04b54f7,
resourceIdName: z30013507-eva-09, domain_name=El_dws_lwx565140,
domain_id=0676610f3a0a4c2c80c50bea7ddf18c1,
res_domain_name=op_svc_dws_0676610f3a0a4c2c80c50bea7ddf18c1",
    "alarm_location": "cluster_id: 5e76e8e2-d0cf-4b64-9d9a-aadbb04b54f7,cluster_name: z30013507-eva-09,
集群类型: dws,domain_name: El_dws_demo,domain_id: 0676610f3a0a4c2c80c50bea7ddf18c1,",
    "resource_id": "5e76e8e2-d0cf-4b64-9d9a-aadbb04b54f7",
    "resource_id_name": "z30013507-eva-09",
    "alarm_generate_date": "2022-10-27 08:11:29",
    "alarm_status": "0"
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAlarmDetailSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAlarmDetailRequest request = new ListAlarmDetailRequest();
        request.withTimezone("<time_zone>");
        request.withOffset("<offset>");
        request.withLimit("<limit>");
        try {
            ListAlarmDetailResponse response = client.listAlarmDetail(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAlarmDetailRequest()
        request.time_zone = "<time_zone>"
        request.offset = "<offset>"
        request.limit = "<limit>"
        response = client.list_alarm_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAlarmDetailRequest{}
request.TimeZone = "<time_zone>"
offsetRequest:= "<offset>"
request.Offset = &offsetRequest
limitRequest:= "<limit>"
request.Limit = &limitRequest
response, err := client.ListAlarmDetail(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	查询告警详情列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.3 查询告警统计列表

功能介绍

该接口用于查询告警统计列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/alarm-statistic

表 5-389 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 5-390 Query 参数

参数	是否必选	参数类型	描述
time_zone	是	String	时区。

请求参数

无

响应参数

状态码： 200

表 5-391 响应 Body 参数

参数	参数类型	描述
alarm_statistics	Array of AlarmStatisticResponse objects	告警统计列表。

表 5-392 AlarmStatisticResponse

参数	参数类型	描述
date	String	日期。
urgent	String	紧急。
important	String	重要。
minor	String	次要。

参数	参数类型	描述
prompt	String	提示。

请求示例

https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarm-statistic?time_zone=GMT

响应示例

状态码： 200

```
{
  "alarm_statistics" : [ {
    "date" : "2022-10-21",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-22",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-23",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-24",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-25",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-26",
    "urgent" : 0,
    "important" : 0,
    "minor" : 0,
    "prompt" : 0
  }, {
    "date" : "2022-10-27",
    "urgent" : 17,
    "important" : 0,
    "minor" : 0,
    "prompt" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAlarmStatisticSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAlarmStatisticRequest request = new ListAlarmStatisticRequest();
        request.withTimeZone("<time_zone>");
        try {
            ListAlarmStatisticResponse response = client.listAlarmStatistic(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListAlarmStatisticRequest()
    request.time_zone = "<time_zone>"
    response = client.list_alarm_statistic(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAlarmStatisticRequest{}
    request.TimeZone = "<time_zone>"
    response, err := client.ListAlarmStatistic(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	查询告警统计列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.4 查询告警订阅列表

功能介绍

该接口用于查询订阅告警。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/alarm-sub

表 5-393 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 5-394 Query 参数

参数	是否必选	参数类型	描述
offset	否	String	当前页。默认取值为1，最大支持10000。
limit	否	String	显示条数。最大支持1000。

请求参数

无

响应参数

状态码： 200

表 5-395 响应 Body 参数

参数	参数类型	描述
count	Integer	告警订阅总数。
alarm_subscriptions	Array of AlarmSubscriptionResponse objects	告警订阅列表。

表 5-396 AlarmSubscriptionResponse

参数	参数类型	描述
id	String	告警订阅ID。
name	String	告警订阅名称。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none"> • urgent: 紧急 • important: 重要 • minor: 次要 • prompt: 提示
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息主题地址。
notification_target_name	String	消息主题名称。
notification_target_type	String	消息主题类型。
language	String	语言。
time_zone	String	时区。

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarm-subs
```

响应示例

状态码： 200

```
{
  "count" : 1,
  "alarm_subscriptions" : [ {
    "id" : "e8d8359f-b8bd-4b80-bc4d-32c86c7c725e",
    "name" : "00",
    "enable" : 1,
    "language" : "zh-cn",
    "alarm_level" : "urgent,important,minor,prompt",
    "project_id" : "4cf650fd46704908aa071b4df2453e1e",
    "name_space" : "DWS",
    "notification_target" : "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:CGS",
    "notification_target_name" : "CGS",
    "notification_target_type" : "SMN",
    "time_zone" : "GMT+08:00"
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAlarmSubsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListAlarmSubsRequest request = new ListAlarmSubsRequest();
        try {
            ListAlarmSubsResponse response = client.listAlarmSubs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListAlarmSubsRequest()
        response = client.list_alarm_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAlarmSubsRequest{}
response, err := client.ListAlarmSubs(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	查询告警订阅列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.5 创建告警订阅

功能介绍

该接口用于创建告警订阅。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/alarm-sub

表 5-397 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 5-398 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	告警订阅名称。
enable	否	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
alarm_level	否	String	告警级别。其中包括： <ul style="list-style-type: none"> • urgent：紧急 • important：重要 • minor：次要 • prompt：提示
notification_target	是	String	消息主题地址。
notification_target_name	是	String	消息主题名称。
notification_target_type	是	String	消息主题类型，支持SMN。
time_zone	是	String	时区。

响应参数

状态码： 200

表 5-399 响应 Body 参数

参数	参数类型	描述
id	String	告警订阅ID。
name	String	告警订阅名称。

参数	参数类型	描述
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none"> • urgent: 紧急 • important: 重要 • minor: 次要 • prompt: 提示
project_id	String	项目ID。
name_space	String	所属服务
notification_target	String	消息主题地址。
notification_target_name	String	消息主题名称。
notification_target_type	String	消息主题类型。
language	String	语言。
time_zone	String	时区。

请求示例

开启名为“zrf-test-12”的紧急、重要、次要告警订阅，并选择地址为“urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete”的SMN消息主题“dws-test-nodelete”：

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarm-sub
```

```
{
  "alarm_level": "urgent,important,minor",
  "enable": 1,
  "name": "zrf-test-12",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

响应示例

状态码： 200

```
{
  "id": "273ce506-dad8-411c-92f9-be5004739b40",
  "name": "zrf-test-12",
  "enable": 1,
  "language": "zh-cn",
}
```

```
"alarm_level": "urgent,important,minor",
"project_id": "4cf650fd46704908aa071b4df2453e1e",
"name_space": "dws",
"notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
"notification_target_name": "dws-test-nodelete",
"notification_target_type": "SMN",
"time_zone": "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateAlarmSubSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CreateAlarmSubRequest request = new CreateAlarmSubRequest();
        AlarmSubRequest body = new AlarmSubRequest();
        body.withTimeZone("GMT+08:00");
        body.withNotificationTargetType("SMN");
        body.withNotificationTargetName("dws-test-nodelete");
        body.withNotificationTarget("urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-
nodelete");
        body.withAlarmLevel("urgent,important,minor");
        body.withEnable(1);
        body.withName("zrf-test-12");
        request.withBody(body);
        try {
            CreateAlarmSubResponse response = client.createAlarmSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = CreateAlarmSubRequest()
        request.body = AlarmSubRequest(
            time_zone="GMT+08:00",
            notification_target_type="SMN",
            notification_target_name="dws-test-nodelete",
            notification_target="urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
            alarm_level="urgent,important,minor",
            enable=1,
            name="zrf-test-12"
        )
        response = client.create_alarm_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.CreateAlarmSubRequest{}
alarmLevelAlarmSubRequest:= "urgent,important,minor"
enableAlarmSubRequest:= int32(1)
request.Body = &model.AlarmSubRequest{
    TimeZone: "GMT+08:00",
    NotificationTargetType: "SMN",
    NotificationTargetName: "dws-test-nodelete",
    NotificationTarget: "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
    AlarmLevel: &alarmLevelAlarmSubRequest,
    Enable: &enableAlarmSubRequest,
    Name: "zrf-test-12",
}
response, err := client.CreateAlarmSub(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	创建告警订阅成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.6 更新告警订阅

功能介绍

该接口用于更新订阅的告警。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/alarm-sub/{alarm_sub_id}

表 5-400 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 。
alarm_sub_id	是	String	告警订阅ID。

请求参数

表 5-401 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	告警订阅名称。
enable	否	Integer	是否开启订阅： <ul style="list-style-type: none"> 1为开启 0为关闭
alarm_level	否	String	告警级别。其中包括： <ul style="list-style-type: none"> urgent：紧急 important：重要 minor：次要 prompt：提示
notification_target	是	String	消息主题地址。
notification_target_name	是	String	消息主题名称。
notification_target_type	是	String	消息主题类型，仅支持SMN。

响应参数

状态码： 200

表 5-402 响应 Body 参数

参数	参数类型	描述
id	String	告警订阅ID。
name	String	告警订阅名称。
enable	Integer	是否开启订阅： <ul style="list-style-type: none"> • 1为开启 • 0为关闭
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none"> • urgent：紧急 • important：重要 • minor：次要 • prompt：提示
project_id	String	项目ID。
name_space	String	所属服务。
notification_target	String	消息主题地址。
notification_target_name	String	消息主题名称。
notification_target_type	String	消息主题类型。
language	String	语言。
time_zone	String	时区。

请求示例

修改zrf-test-13告警订阅，告警级别修改为紧急、重要、次要，SMN消息主题名称修改为dws-test-nodelete，地址修改为“urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete”：

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarm-sub/273ce506-dad8-411c-92f9-be5004739b40
```

```
{
  "alarm_level": "urgent,important,minor",
  "enable": 1,
  "name": "zrf-test-13",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN"
}
```

响应示例

状态码： 200

```
{
  "id": "273ce506-dad8-411c-92f9-be5004739b40",
  "name": "zrf-test-13",
  "enable": 1,
  "language": "zh-cn",
  "alarm_level": "urgent,important,minor",
  "project_id": "4cf650fd46704908aa071b4df2453e1e",
  "name_space": "dws",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateAlarmSubSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        UpdateAlarmSubRequest request = new UpdateAlarmSubRequest();
        AlarmSubUpdateRequest body = new AlarmSubUpdateRequest();
        body.withNotificationTargetType("SMN");
        body.withNotificationTargetName("dws-test-nodelete");
        body.withNotificationTarget("urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-
nodelete");
        body.withAlarmLevel("urgent,important,minor");
        body.withEnable(1);
        body.withName("zrf-test-13");
        request.withBody(body);
        try {
            UpdateAlarmSubResponse response = client.updateAlarmSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateAlarmSubRequest()
        request.body = AlarmSubUpdateRequest(
            notification_target_type="SMN",
            notification_target_name="dws-test-nodelete",
            notification_target="urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
            alarm_level="urgent,important,minor",
            enable=1,
            name="zrf-test-13"
        )
        response = client.update_alarm_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.UpdateAlarmSubRequest{}
alarmLevelAlarmSubUpdateRequest:= "urgent,important,minor"
enableAlarmSubUpdateRequest:= int32(1)
request.Body = &model.AlarmSubUpdateRequest{
    NotificationTargetType: "SMN",
    NotificationTargetName: "dws-test-nodelete",
    NotificationTarget: "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
    AlarmLevel: &alarmLevelAlarmSubUpdateRequest,
    Enable: &enableAlarmSubUpdateRequest,
    Name: "zrf-test-13",
}
response, err := client.UpdateAlarmSub(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	更新告警订阅成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.8.7 删除告警订阅

功能介绍

该接口用于删除订阅的告警。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/alarm-sub/{alarm_sub_id}

表 5-403 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 。
alarm_sub_id	是	String	告警订阅ID。

请求参数

无

响应参数

状态码： 200

表 5-404 响应 Body 参数

参数	参数类型	描述
id	String	告警订阅ID。
name	String	告警订阅名称。
enable	Integer	是否开启订阅： <ul style="list-style-type: none">• 1为开启• 0为关闭
alarm_level	String	告警级别。其中包括： <ul style="list-style-type: none">• urgent: 紧急• important: 重要• minor: 次要• prompt: 提示
project_id	String	项目ID。

参数	参数类型	描述
name_space	String	所属服务。
notification_target	String	消息主题地址。
notification_target_name	String	消息主题名称。
notification_target_type	String	消息主题类型。
language	String	语言。
time_zone	String	时区。

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/alarm-subs/273ce506-dad8-411c-92f9-be5004739b40
```

响应示例

状态码： 200

```
{
  "id": "273ce506-dad8-411c-92f9-be5004739b40",
  "name": "zrf-test-13",
  "enable": 1,
  "language": "zh-cn",
  "alarm_level": "urgent,important,minor",
  "project_id": "4cf650fd46704908aa071b4df2453e1e",
  "name_space": "dws",
  "notification_target": "urn:smn:cn-north-7:4cf650fd46704908aa071b4df2453e1e:dws-test-nodelete",
  "notification_target_name": "dws-test-nodelete",
  "notification_target_type": "SMN",
  "time_zone": "GMT+08:00"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteAlarmSubSolution {
    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
DeleteAlarmSubRequest request = new DeleteAlarmSubRequest();
try {
    DeleteAlarmSubResponse response = client.deleteAlarmSub(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = DeleteAlarmSubRequest()
        response = client.delete_alarm_sub(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAlarmSubRequest{}
    response, err := client.DeleteAlarmSub(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	删除告警订阅成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9 连接管理

5.9.1 申请域名

功能介绍

该接口用于为指定集群申请域名。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/dns

表 5-405 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-406 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	待创建的域名。 说明 域名可以包含中划线-，但不允许包含下划线_。
type	是	String	域名类型。 <ul style="list-style-type: none"> public: 公网域名。 private: 内网域名。
ttd	是	Integer	用于填写默认生成的SOA记录中有效缓存时间，以秒为单位。 <ul style="list-style-type: none"> 取值范围：300~2147483647。 默认值为300s。

响应参数

无

请求示例

申请公网域名demo，默认生成的SOA记录中有效缓存时间为300s：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/dns
{
  "name" : "demo",
  "type" : "public",
  "ttl" : 300
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateClusterDnsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateClusterDnsRequest request = new CreateClusterDnsRequest();
        CreateClusterDns body = new CreateClusterDns();
        body.withTtl(300);
        body.withType("public");
        body.withName("demo");
        request.withBody(body);
        try {
```

```
        CreateClusterDnsResponse response = client.createClusterDns(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateClusterDnsRequest()
        request.body = CreateClusterDns(
            ttl=300,
            type="public",
            name="demo"
        )
        response = client.create_cluster_dns(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateClusterDnsRequest{}
request.Body = &model.CreateClusterDns{
    Ttl: int32(300),
    Type: "public",
    Name: "demo",
}
response, err := client.CreateClusterDns(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	申请域名成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.2 修改集群域名

功能介绍

该接口用于为指定集群修改域名。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1.0/{project_id}/clusters/{cluster_id}/dns

表 5-407 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-408 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	待修改的域名。
type	是	String	域名类型。 <ul style="list-style-type: none"> public: 公网域名。 private: 内网域名。
ttd	是	Integer	用于填写默认生成的SOA记录中有效缓存时间，以秒为单位。 <ul style="list-style-type: none"> 取值范围：300~2147483647。 默认值为300s。

响应参数

无

请求示例

修改集群公网域名demo默认生成的SOA记录中有效缓存时间为300s:

```
PUT https://{Endpoint}/v1.0/89cd04f168b84af6be287f1730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/dns
{
  "name": "demo",
  "type": "public",
  "ttl": 300
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateClusterDnsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateClusterDnsRequest request = new UpdateClusterDnsRequest();
        ModifyClusterDns body = new ModifyClusterDns();
        body.withTtl(300);
        body.withType("public");
        body.withName("demo");
        request.withBody(body);
        try {
            UpdateClusterDnsResponse response = client.updateClusterDns(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
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateClusterDnsRequest()
        request.body = ModifyClusterDns(
            ttl=300,
            type="public",
            name="demo"
        )
        response = client.update_cluster_dns(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

```
request.Body = &model.ModifyClusterDns{
    Ttl: int32(300),
    Type: "public",
    Name: "demo",
}
response, err := client.UpdateClusterDns(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	修改集群域名成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.3 删除集群域名

功能介绍

该接口用于删除指定集群域名。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v1.0/{project_id}/clusters/{cluster_id}/dns
```

表 5-409 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

表 5-410 Query 参数

参数	是否必选	参数类型	描述
type	是	String	域名类型，支持删除公网域名。

请求参数

无

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/dns?type=public
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteClusterDnsSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteClusterDnsRequest request = new DeleteClusterDnsRequest();
request.withType("<type>");
try {
    DeleteClusterDnsResponse response = client.deleteClusterDns(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteClusterDnsRequest()
        request.type = "<type>"
        response = client.delete_cluster_dns(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteClusterDnsRequest{}
    request.Type = "<type>"
    response, err := client.DeleteClusterDns(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	删除集群域名成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.4 获取集群可绑定的 ELB 列表

功能介绍

该接口用于查询集群可以关联的ELB列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/clusters/{cluster_id}/elbs

表 5-411 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-412 响应 Body 参数

参数	参数类型	描述
elbs	Array of ClusterElbInfo objects	弹性负载均衡列表。

表 5-413 ClusterElbInfo

参数	参数类型	描述
id	String	弹性负载均衡ID。
cluster_id	String	集群ID。
name	String	弹性负载均衡名称。

参数	参数类型	描述
description	String	弹性负载均衡描述。
vip_address	String	弹性负载均衡地址。
vip_subnet_id	String	子网ID。
tenant_id	String	租户ID。
type	String	弹性负载均衡类型。其中包括： <ul style="list-style-type: none"> Internal：独享型。 External：共享型。
admin_state_up	Boolean	弹性负载均衡的管理状态。其中包括： <ul style="list-style-type: none"> ACTIVE：使用中。 PENDING_CREATE：创建中。 ERROR：表示当前策略与同一监听器下的其他策略存在相同的规则配置。
bandwidth	Integer	绑定状态： <ul style="list-style-type: none"> 0：未绑定 1：已绑定
vpc_id	String	虚拟私有云ID。

请求示例

```
GET https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/clusters/194408fa-9d41-435c-a140-91edcf5fe519/elbs
```

响应示例

状态码： 200

弹性负载均衡列表

```
{
  "elbs": [ {
    "id": "1e6e0b66-6223-4523-bfd9-033c88b4ce9f",
    "name": "loadbalancer5",
    "description": "simple lb",
    "bandwidth": 0,
    "vip_address": "192.168.0.222",
    "admin_state_up": true,
    "vpc_id": "c9f1171e-dc90-4ae9-bf22-f9736983ce2d"
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListElbsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListElbsRequest request = new ListElbsRequest();
        try {
            ListElbsResponse response = client.listElbs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListElbsRequest()
    response = client.list_elbs(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListElbsRequest{}
    response, err := client.ListElbs(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	获取弹性负载均衡列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.5 集群绑定 ELB

功能介绍

该接口用于集群绑定ELB接口。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/elbs/{elb_id}

表 5-414 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
elb_id	是	String	未绑定的弹性负载均衡ID。

请求参数

无

响应参数

状态码： 200

表 5-415 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/clusters/194408fa-9d41-435c-a140-91edcf5fe519/elbs/1e6e0b66-6223-4523-bfd9-033c88b4ce9f
```

响应示例

状态码： 200

```
{"job_id":"2c9081838417d8850184196d8282002b"}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class AssociateElbSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        AssociateElbRequest request = new AssociateElbRequest();
        try {
            AssociateElbResponse response = client.associateElb(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AssociateElbRequest()
        response = client.associate_elb(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
    dws.DwsClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.AssociateElbRequest{}  
response, err := client.AssociateElb(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	集群绑定ELB成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.6 集群解绑 ELB

功能介绍

该接口用于集群解绑ELB。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/clusters/{cluster_id}/elbs/{elb_id}
```


表 5-416 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
elb_id	是	String	集群已绑定的弹性负载均衡ID。

请求参数

无

响应参数

状态码： 200

表 5-417 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/clusters/194408fa-9d41-435c-a140-91edcf5fe519/elbs/1e6e0b66-6223-4523-bfd9-033c88b4ce9f
```

响应示例

状态码： 200

```
{"job_id":"2c9081838417d8850184196d8282002b"}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class DisassociateElbSolution {  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        DisassociateElbRequest request = new DisassociateElbRequest();  
        try {  
            DisassociateElbResponse response = client.disassociateElb(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = DisassociateElbRequest()  
        response = client.disassociate_elb(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisassociateElbRequest{}
    response, err := client.DisassociateElb(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	集群解绑ELB成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.9.7 集群绑定 EIP

功能介绍

该接口用于集群绑定EIP。

说明

默认情况下，只有华为云账号用户或拥有Security Administrator权限的用户才具备查询委托和创建委托的权限。账号中的IAM用户，默认没有查询委托和创建委托的权限，在使用弹性IP绑定功能时页面会屏蔽绑定按钮，此时需联系有“DWS Administrator”权限的用户在当前页面完成对GaussDB(DWS)的委托授权，详情请参见[委托GaussDB\(DWS\)管理资源](#)。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/eips/{eip_id}

表 5-418 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
eip_id	是	String	未绑定的弹性IP的ID。

请求参数

无

响应参数

无

请求示例

https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/clusters/194408fa-9d41-435c-a140-91edcf5fe519/eips/ab60b4ac-10e3-4d83-bccd-9a6a1b0ba983

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class AssociateEipSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        AssociateEipRequest request = new AssociateEipRequest();
        try {
            AssociateEipResponse response = client.associateEip(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AssociateEipRequest()
        response = client.associate_eip(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateEipRequest{}
    response, err := client.AssociateEip(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	集群绑定EIP成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.9.8 集群解绑 EIP

功能介绍

该接口用于集群解绑EIP。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/clusters/{cluster_id}/eips/{eip_id}
```

表 5-419 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
eip_id	是	String	集群绑定的弹性IP。

请求参数

无

响应参数

无

请求示例

```
https://{Endpoint}/v2/4cf650fd46704908aa071b4df2453e1e/clusters/194408fa-9d41-435c-a140-91edcf5fe519/eips/ab60b4ac-10e3-4d83-bccd-9a6a1b0ba983
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DisassociateEipSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        DisassociateEipRequest request = new DisassociateEipRequest();
        try {
            DisassociateEipResponse response = client.disassociateEip(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = DisassociateEipRequest()  
        response = client.disassociate_eip(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"  
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
        dws.DwsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

```
request := &model.DisassociateEipRequest{}
response, err := client.DisassociateEip(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	集群解绑EIP成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.10 标签管理

5.10.1 查询项目标签

功能介绍

该接口用于查询项目标签列表。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/tags
```

表 5-420 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

无

响应参数

状态码： 200

表 5-421 响应 Body 参数

参数	参数类型	描述
tags	Array of ProjectTag objects	标签列表对象。

表 5-422 ProjectTag

参数	参数类型	描述
key	String	键。输入标签键的最大长度为128个字符，不能为空字符串，且首尾字符不能为空格。不能包含“=”、“*”、“<”、“>”、“\”、“”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。
values	Array of strings	值。输入标签值的最大长度为255个字符，首尾字符不能为空格，可以为空字符串。不能包含“=”、“*”、“<”、“>”、“\”、“”、“ ”、“/”。只能包含大写字母（A-Z）、小写字母（a-z）、数字（0-9）和特殊字符（中划线-、下划线_）以及中文字符。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/tags
```

响应示例

状态码： 200

查询项目标签成功。

```
{
  "tags": [ {
    "key": "key",
    "values": [ "value-1", "value-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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListTagsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTagsRequest request = new ListTagsRequest();
        try {
            ListTagsResponse response = client.listTags(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTagsRequest()
        response = client.list_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTagsRequest{}
    response, err := client.ListTags(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	查询项目标签成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.10.2 查询集群标签

功能介绍

该接口用于查询指定集群的标签信息。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/clusters/{cluster_id}/tags
```

表 5-423 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-424 响应 Body 参数

参数	参数类型	描述
tags	Array of ResourceTag objects	标签列表。

表 5-425 ResourceTag

参数	参数类型	描述
key	String	标签键。
value	String	标签值。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/tags
```

响应示例

状态码： 200

查询集群标签成功。

```
{
  "tags": [ {
    "key": "key",
    "value": "value"
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListClusterTagsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListClusterTagsRequest request = new ListClusterTagsRequest();
        try {
            ListClusterTagsResponse response = client.listClusterTags(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListClusterTagsRequest()
```



```
response = client.list_cluster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClusterTagsRequest{}
    response, err := client.ListClusterTags(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	查询集群标签成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。

状态码	描述
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.10.3 批量添加标签

功能介绍

该接口用于为指定集群批量添加标签。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/tags/batch-create

表 5-426 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-427 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of BatchCreateResourceTag objects	标签列表。

表 5-428 BatchCreateResourceTag

参数	是否必选	参数类型	描述
key	是	String	标签键。 最小长度：1 最大长度：128
value	是	String	标签值。 最小长度：0 最大长度：255

响应参数

无

请求示例

批量添加键为key，值为value的标签：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/tags/batch-create
{
  "tags": [
    {
      "key": "key",
      "value": "value"
    }
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class BatchCreateResourceTagSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
BatchCreateResourceTagRequest request = new BatchCreateResourceTagRequest();
BatchCreateResourceTags body = new BatchCreateResourceTags();
List<BatchCreateResourceTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new BatchCreateResourceTag()
        .withKey("key")
        .withValue("value")
);
body.withTags(listbodyTags);
request.withBody(body);
try {
    BatchCreateResourceTagResponse response = client.batchCreateResourceTag(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
```

```
request = BatchCreateResourceTagRequest()
listTagsbody = [
    BatchCreateResourceTag(
        key="key",
        value="value"
    )
]
request.body = BatchCreateResourceTags(
    tags=listTagsbody
)
response = client.batch_create_resource_tag(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagRequest{}
    var listTagsbody = []model.BatchCreateResourceTag{
        {
            Key: "key",
            Value: "value",
        },
    }
    request.Body = &model.BatchCreateResourceTags{
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateResourceTag(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	批量添加标签成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.10.4 批量删除标签

功能介绍

该接口用于为指定集群批量删除标签。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters/{cluster_id}/tags/batch-delete

表 5-429 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-430 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of BatchDeleteResourceTag objects	标签列表。

表 5-431 BatchDeleteResourceTag

参数	是否必选	参数类型	描述
key	是	String	标签键。
value	是	String	标签值。

响应参数

无

请求示例

批量删除键为key，值为value的标签：

```
POST https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7/tags/batch-delete
{
  "tags": [
    {
      "key": "key",
      "value": "value"
    }
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class BatchDeleteResourceTagSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchDeleteResourceTagRequest request = new BatchDeleteResourceTagRequest();
        BatchDeleteResourceTags body = new BatchDeleteResourceTags();
        List<BatchDeleteResourceTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new BatchDeleteResourceTag()
                .withKey("key")
                .withValue("value")
        );
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchDeleteResourceTagResponse response = client.batchDeleteResourceTag(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = BatchDeleteResourceTagRequest()
    listTagsbody = [
        BatchDeleteResourceTag(
            key="key",
            value="value"
        )
    ]
    request.body = BatchDeleteResourceTags(
        tags=listTagsbody
    )
    response = client.batch_delete_resource_tag(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteResourceTagRequest{}
    var listTagsbody = []model.BatchDeleteResourceTag{
        {
            Key: "key",
            Value: "value",
        },
    }
    request.Body = &model.BatchDeleteResourceTags{
        Tags: listTagsbody,
    }
}
```

```
}  
response, err := client.BatchDeleteResourceTag(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	批量删除标签成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.11 配额管理

5.11.1 查询配额

功能介绍

该接口用于查询单租户在GaussDB(DWS)服务下的配额信息。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/quotas
```

表 5-432 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

无

响应参数

状态码： 200

表 5-433 响应 Body 参数

参数	参数类型	描述
quotas	QuotasResources object	配额列表对象。

表 5-434 QuotasResources

参数	参数类型	描述
resources	Array of QuotasResource objects	资源列表对象。

表 5-435 QuotasResource

参数	参数类型	描述
type	String	项目资源类型。
used	String	已使用的资源数量。
quota	Integer	项目资源配额。
unit	Integer	资源计量单位。

请求示例

```
GET https://{Endpoint}/v1.0/89cd04f168b84af6be287f71730fdb4b/quotas
```

响应示例

状态码： 200

查询配额成功。

```
{
  "quotas": {
    "resources": [ {
      "type": "instances",
      "used": "3",
      "quota": "100",
      "unit": "null"
    }, {
      "type": "core",
      "used": "12",
      "quota": "100",
      "unit": "null"
    }, {
      "type": "ram",
      "used": "24",
      "quota": "1000",
      "unit": "GB"
    }, {
      "type": "disk",
      "used": "6",
      "quota": "100",
      "unit": "null"
    }, {
      "type": "disk_capacity",
      "used": "600",
      "quota": "1000",
      "unit": "GB"
    }
  ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListQuotasSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
        .build();
ListQuotasRequest request = new ListQuotasRequest();
try {
    ListQuotasResponse response = client.listQuotas(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListQuotasRequest()
        response = client.list_quotas(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListQuotasRequest{}
response, err := client.ListQuotas(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	查询配额成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12 容灾管理

5.12.1 创建容灾

功能介绍

该接口用于创建容灾。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/disaster-recoveries

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 。

请求参数

表 5-437 请求 Body 参数

参数	是否必选	参数类型	描述
disaster_recovery	是	DisasterRecoveryReq object	容灾对象。

表 5-438 DisasterRecoveryReq

参数	是否必选	参数类型	描述
name	是	String	名称。容灾名称在4位到64位之间，不区分大小写，必须以字母开头，可以包含字母、数字、中划线或者下划线，不能包含其他的特殊字符。
dr_type	是	String	容灾类型，其中包含：az。
dr_sync_period	是	String	同步周期，范围为1-3000。 单位： <ul style="list-style-type: none"> 分钟 m 小时 H 天 d
primary_cluster_id	是	String	主集群ID。
standby_cluster_id	是	String	备集群ID。

参数	是否必选	参数类型	描述
primary_obs_bucket	否	String	主集群obs桶。
standby_obs_bucket	否	String	备集群obs桶。

响应参数

表 5-439 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecovery object	容灾对象。

表 5-440 DisasterRecovery

参数	参数类型	描述
id	String	容灾ID。

请求示例

创建名为test_dr4的跨AZ容灾，设置每5分钟同步一次数据，其中主集群ID为“3ef5daaa-d017-4354-80b3-8a69caf2e09”，备集群ID为“e7fed7fe-f4a7-499a-afed-1fdb5c1978cb”：

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recoveries
{
  "disaster_recovery": {
    "name": "test_dr4",
    "dr_type": "az",
    "primary_cluster_id": "3ef5daaa-d017-4354-80b3-8a69caf2e094",
    "standby_cluster_id": "e7fed7fe-f4a7-499a-afed-1fdb5c1978cb",
    "dr_sync_period": "5m",
    "primary_obs_bucket": "",
    "standby_obs_bucket": ""
  }
}
```

响应示例

状态码： 200

创建容灾成功

```
{
  "disaster_recovery": {
    "id": "aa1a688b-4bde-44ec-931f-bd77042841fb"
  }
}
```


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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CreateDisasterRecoverySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CreateDisasterRecoveryRequest request = new CreateDisasterRecoveryRequest();
        CreateDisasterRecoveryReq body = new CreateDisasterRecoveryReq();
        request.withBody(body);
        try {
            CreateDisasterRecoveryResponse response = client.createDisasterRecovery(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = CreateDisasterRecoveryRequest()
    request.body = CreateDisasterRecoveryReq(
    )
    response = client.create_disaster_recovery(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CreateDisasterRecoveryRequest{}
    request.Body = &model.CreateDisasterRecoveryReq{
    }
    response, err := client.CreateDisasterRecovery(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	创建容灾成功
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.2 启动容灾

功能介绍

该接口用于启动容灾操作。容灾状态为“未启动”、“启动失败”和“已停止”时可以执行启动容灾操作。启动容灾后，生产集群和灾备集群将无法进行恢复、扩容、升级、重启、节点替换、更新密码等操作，此外，灾备集群将无法进行备份操作，请谨慎操作。当容灾启动后，如果灾备集群容灾正常运行且容灾处于恢复状态中，此状态的集群会计费。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/disaster-recovery/{disaster_recovery_id}/start

表 5-441 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-442 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-443 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/e59d6b86-9072-46eb-a996-13f8b44994c1/start
```

响应示例

状态码： 200

启动容灾下发成功。

```
{
  "disaster_recovery" : {
    "id" : "e59d6b86-9072-46eb-a996-13f8b44994c1"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class StartDisasterRecoverySolution {
    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        StartDisasterRecoveryRequest request = new StartDisasterRecoveryRequest();
        try {
            StartDisasterRecoveryResponse response = client.startDisasterRecovery(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = StartDisasterRecoveryRequest()
        response = client.start_disaster_recovery(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StartDisasterRecoveryRequest{}
    response, err := client.StartDisasterRecovery(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	启动容灾下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.12.3 查询容灾列表

功能介绍

该接口用于查询容灾列表。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/disaster-recoveries

表 5-444 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

无

响应参数

状态码： 200

表 5-445 响应 Body 参数

参数	参数类型	描述
disaster_recovery	Array of DisasterRecovery objects	容灾列表。

表 5-446 DisasterRecovery

参数	参数类型	描述
id	String	容灾ID。

参数	参数类型	描述
name	String	容灾名称。
dr_type	String	容灾类型。
status	String	容灾状态： <ul style="list-style-type: none"> ● creating：创建中。 ● create_failed：创建失败。 ● unstart：未启动。 ● starting：启动中。 ● start_failed：启动失败。 ● running：运行中。 ● stopping：停止中。 ● stop_failed：停止失败。 ● stopped：已停止。 ● switchovering：灾备切换中。 ● abnormal：异常。 ● deleting：删除中。
primary_cluster_id	String	主集群ID。
primary_cluster_name	String	主集群名称。
standby_cluster_id	String	备集群ID。
standby_cluster_name	String	备集群名称。
primary_cluster_role	String	主集群角色。
standby_cluster_role	String	备集群角色。
primary_cluster_status	String	主集群状态： <ul style="list-style-type: none"> ● backuping：备份中。 ● stopped：已停止。 ● waiting：等待下个周期执行中。 ● abnormal：异常。

参数	参数类型	描述
standby_cluster_status	String	备集群状态： <ul style="list-style-type: none"> restoring：恢复中。 stopped：已停止。 waiting：等待下个周期执行中。 abnormal：异常。
primary_cluster_region	String	主集群Region。
standby_cluster_region	String	备集群Region。
primary_cluster_project_id	String	主集群项目ID。
standby_cluster_project_id	String	备集群项目ID。
last_disaster_time	String	最后一次容灾同步时间。
start_time	String	启动时间。
create_time	String	创建时间。

请求示例

GET https://{Endpoint}/v2/89cd04f168b84af6be287f1730fdb4b/disaster-recoveries

响应示例

```
{
  "disaster_recovery": [
    {
      "id": "9956ee8a-2f5b-4d76-8c18-4115e2393433",
      "name": "test-dr-1026-3",
      "status": "stop_failed",
      "dr_type": "az",
      "primary_cluster_id": "f6dc1f82-8d66-4750-a5d2-569d3f72cf61",
      "primary_cluster_name": "dr_2",
      "standby_cluster_id": "fb34f3f3-f86a-446b-9f2f-e0581ba55e73",
      "standby_cluster_name": "dr_1",
      "primary_cluster_role": "primary",
      "standby_cluster_role": "standby",
      "primary_cluster_status": "waiting",
      "standby_cluster_status": "waiting",
      "primary_cluster_region": "cn-north-7",
      "standby_cluster_region": "cn-north-7",
      "primary_cluster_project_id": "0536cdee2200d5912f7cc00b877980f1",
      "standby_cluster_project_id": "0536cdee2200d5912f7cc00b877980f1",
      "last_disaster_time": "2022-10-28T02:34:16",
      "start_time": null,
      "create_time": "2022-10-26T12:04:10"
    },
    {
      "id": "a85dee93-cd49-4c52-a44b-c50b24f6caae",
      "name": "test_dis",

```

```
"status": "creating",
"dr_type": "az",
"primary_cluster_id": "3ef5daaa-d017-4354-80b3-8a69caf2e094",
"primary_cluster_name": "t1031_02",
"standby_cluster_id": "a07cb2f7-b17e-4d95-923b-a33d0c884d37",
"standby_cluster_name": "t1028_02",
"primary_cluster_role": "primary",
"standby_cluster_role": "standby",
"primary_cluster_status": null,
"standby_cluster_status": null,
"primary_cluster_region": "cn-north-7",
"standby_cluster_region": "cn-north-7",
"primary_cluster_project_id": "0536cdee2200d5912f7cc00b877980f1",
"standby_cluster_project_id": "0536cdee2200d5912f7cc00b877980f1",
"last_disaster_time": null,
"start_time": null,
"create_time": "2022-10-31T06:17:00"
}
]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListDisasterRecoverSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDisasterRecoverRequest request = new ListDisasterRecoverRequest();
        try {
            ListDisasterRecoverResponse response = client.listDisasterRecover(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDisasterRecoverRequest()
        response = client.list_disaster_recover(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
    dws.DwsClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ListDisasterRecoverRequest{}  
response, err := client.ListDisasterRecover(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	查询容灾列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.4 查询容灾详情

功能介绍

该接口用于查询单个容灾详情。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v2/{project_id}/disaster-recovery/{disaster_recovery_id}
```

表 5-447 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-448 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryQueryResponse object	查询容灾信息返回体。

表 5-449 DisasterRecoveryQueryResp

参数	参数类型	描述
id	String	容灾ID。
name	String	容灾名称。
dr_type	String	容灾类型。

参数	参数类型	描述
status	String	容灾状态： <ul style="list-style-type: none"> • creating：创建中。 • create_failed：创建失败。 • unstart：未启动。 • starting：启动中。 • start_failed：启动失败。 • running：运行中。 • stopping：停止中。 • stop_failed：停止失败。 • stopped：已停止。 • switchovering：灾备切换中。 • abnormal：异常。 • deleting：删除中。
primary_cluster	DisasterRecoveryCluster object	主集群信息。
standby_cluster	DisasterRecoveryCluster object	备集群信息。
dr_sync_period	String	容灾同步周期。
start_time	String	容灾启动时间。
create_time	String	容灾创建时间。

表 5-450 DisasterRecoveryCluster

参数	参数类型	描述
id	String	容灾集群信息ID。
name	String	容灾集群名称。
cluster_az	String	容灾集群所在AZ。
role	String	容灾集群角色。
region	String	容灾集群所在Region。

参数	参数类型	描述
status	String	容灾集群状态： <ul style="list-style-type: none"> restoring：恢复中。 stopped：已停止。 waiting：等待下个周期执行中。 abnormal：异常。
progress	String	容灾进度。
last_success_time	String	上一次容灾时间。
obs_bucket_name	String	OBS桶名称。

请求示例

```
https://{Endpoint}/v2/0536cdee2200d5912f7cc00b877980f1/disaster-recovery/b303b78d-438c-41dd-aaf6-c98eea1fc2a6
```

响应示例

状态码： 200

容灾信息查询成功。

```
{
  "disaster_recovery": {
    "id": "b303b78d-438c-41dd-aaf6-c98eea1fc2a6",
    "name": "dws-it-02",
    "status": "running",
    "dr_type": "region",
    "primary_cluster": {
      "id": "3de21046-e65e-4fcf-86bf-583115b39c27",
      "name": "dws-203-primary",
      "cluster_az": "cn-north-7c",
      "role": "primary",
      "region": "cn-north-7",
      "status": "waiting",
      "progress": "100",
      "last_success_time": "2023-04-29T06:39:56",
      "obs_bucket_name": null
    },
    "standby_cluster": {
      "id": "5d946c51-9ffa-49e3-9129-2bea4d23d3fd",
      "name": "dws-205-standby",
      "cluster_az": null,
      "role": "standby",
      "region": "cn-north-208",
      "status": "waiting",
      "progress": "100",
      "last_success_time": "2023-04-29T06:39:56",
      "obs_bucket_name": null
    },
    "dr_sync_period": "3H",
    "start_time": "2023-04-28T09:23:38",
    "create_time": "2023-04-28T08:51:54"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowDisasterDetailSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ShowDisasterDetailRequest request = new ShowDisasterDetailRequest();
        try {
            ShowDisasterDetailResponse response = client.showDisasterDetail(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = ShowDisasterDetailRequest()
    response = client.show_disaster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowDisasterDetailRequest{}
    response, err := client.ShowDisasterDetail(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	容灾信息查询成功。
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.5 查询容灾恢复进度详情

功能介绍

该接口用于查询容灾进度详情信息操作（本接口指标值仅供参考，实际值需要结合业务运行情况分析）。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1/{project_id}/disaster-recovery/{disaster_recovery_id}/show-progress
```

表 5-451 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

表 5-452 响应 Body 参数

参数	参数类型	描述
disaster_recovery_progress	ClusterDisasterRecovery Object	集群容灾恢复详情。

表 5-453 ClusterDisasterRecovery

参数	参数类型	描述
primary_cluster	ClusterRecoveryProgress Object	主集群容灾进度详情。
standby_cluster	ClusterRecoveryProgress Object	备集群容灾进度详情。
latest_barrier_time	String	灾难发生时间。
last_recovery_spend	Long	上一个备份集恢复消耗时间，单位：s。
recovery_point_object	Long	数据恢复目标时间，单位：s。
recovery_time_object	Long	服务恢复目标时间，单位：s。

表 5-454 ClusterRecoveryProgress

参数	参数类型	描述
key	String	本次备份恢复ID。
action_type	String	动作类型。
unrestore_keys	String	待恢复的备份集ID。
action_start_time	String	当前动作开始时间。
action_end_time	String	当前动作结束时间。

请求示例

```
GET https://{Endpoint}/v1/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/  
2a4d0f86-67cd-408a-8b66-017454fb7793/show-progress
```

响应示例

状态码：200

查询集群容灾恢复详情成功。

```
{  
  "disaster_recovery_progress":{  
    "primary_cluster":{  
      "key": "20230327_201637",  
      "action_type": "Backup",  
      "unrestore_keys": "N/A",  
      "action_start_time": "2023-03-27 20:16:34",  
      "action_end_time": "2023-03-27 20:17:00"  
    },  
    "standby_cluster": {  
      "key": "20230327_201637",  
      "action_type": "Restore",  
      "unrestore_keys": "20230327_201637,20230327_201638,20230327_201639",  
      "action_start_time": "2023-03-27 20:18:27",  
      "action_end_time": "2023-03-27 20:24:40"  
    },  
    "latest_barrier_time": "2023-03-28 10:24:40",  
    "last_recovery_spend": 30,  
    "recovery_point_object": 50886,  
    "recovery_time_object": 600  
  }  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ShowDisasterProgressSolution {  
  
    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);
```

```
DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ShowDisasterProgressRequest request = new ShowDisasterProgressRequest();
try {
    ShowDisasterProgressResponse response = client.showDisasterProgress(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowDisasterProgressRequest()
        response = client.show_disaster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ShowDisasterProgressRequest{}
    response, err := client.ShowDisasterProgress(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	查询容灾列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.6 查询可用容灾集群列表

功能介绍

该接口用于查询可用容灾集群列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/disaster-recovery-clusters

表 5-455 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 5-456 Query 参数

参数	是否必选	参数类型	描述
primary_cluster_id	是	String	主集群ID。
primary_spec_id	否	String	主集群规格ID。
primary_cluster_dn_num	否	String	主集群DN数量。
standby_region	否	String	备集群所在Region。
standby_project_id	否	String	备集群项目ID。
standby_az_code	是	String	备集群所在AZ。
dr_type	否	String	容灾类型。
datastore_type	否	String	数仓类型。
datastore_version	否	String	数仓版本。

请求参数

无

响应参数

状态码： 200

表 5-457 响应 Body 参数

参数	参数类型	描述
disaster_recovery_clusters	Array of DisasterRecoveryClusterVo objects	容灾可用集群列表。

表 5-458 DisasterRecoveryClusterVo

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。

请求示例

```
https://{Endpoint}/v2/0536cdee2200d5912f7cc00b877980f1/disaster-recovery-clusters?primary_cluster_id=3de21046-e65e-4fcf-86bf-583115b39c27&standby_az_code=cn-north-7a&standby_region=&dr_type=&standby_project_id=
```

响应示例

状态码： 200

请求成功！

```
{
  "disaster_recovery_clusters": []
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListAvailableDisasterClustersSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ListAvailableDisasterClustersRequest request = new ListAvailableDisasterClustersRequest();
request.withPrimaryClusterId("<primary_cluster_id>");
request.withPrimarySpecId("<primary_spec_id>");
request.withPrimaryClusterDnNum("<primary_cluster_dn_num>");
request.withStandbyRegion("<standby_region>");
request.withStandbyProjectId("<standby_project_id>");
request.withStandbyAzCode("<standby_az_code>");
request.withDrType("<dr_type>");
request.withDatastoreType("<datastore_type>");
request.withDatastoreVersion("<datastore_version>");
try {
    ListAvailableDisasterClustersResponse response = client.listAvailableDisasterClusters(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListAvailableDisasterClustersRequest()
        request.primary_cluster_id = "<primary_cluster_id>"
        request.primary_spec_id = "<primary_spec_id>"
        request.primary_cluster_dn_num = "<primary_cluster_dn_num>"
```

```
request.standby_region = "<standby_region>"
request.standby_project_id = "<standby_project_id>"
request.standby_az_code = "<standby_az_code>"
request.dr_type = "<dr_type>"
request.datastore_type = "<datastore_type>"
request.datastore_version = "<datastore_version>"
response = client.list_available_disaster_clusters(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListAvailableDisasterClustersRequest{}
    request.PrimaryClusterId = "<primary_cluster_id>"
    primarySpecIdRequest := "<primary_spec_id>"
    request.PrimarySpecId = &primarySpecIdRequest
    primaryClusterDnNumRequest := "<primary_cluster_dn_num>"
    request.PrimaryClusterDnNum = &primaryClusterDnNumRequest
    standbyRegionRequest := "<standby_region>"
    request.StandbyRegion = &standbyRegionRequest
    standbyProjectIdRequest := "<standby_project_id>"
    request.StandbyProjectId = &standbyProjectIdRequest
    request.StandbyAzCode = "<standby_az_code>"
    drTypeRequest := "<dr_type>"
    request.DrType = &drTypeRequest
    datastoreTypeRequest := "<datastore_type>"
    request.DatastoreType = &datastoreTypeRequest
    datastoreVersionRequest := "<datastore_version>"
    request.DatastoreVersion = &datastoreVersionRequest
    response, err := client.ListAvailableDisasterClusters(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	请求成功!
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.7 更新容灾配置

功能介绍

该接口用于更新容灾配置操作。容灾状态为“未启动”或“已停止”时，可以执行容灾配置修改操作。新的配置在容灾重新启动后生效。

调用方法

请参见[如何调用API](#)。

URI

```
PUT /v2/{project_id}/disaster-recovery/{disaster_recovery_id}
```

表 5-459 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

表 5-460 请求 Body 参数

参数	是否必选	参数类型	描述
disaster_recovery	否	UpdateDisasterRecoveryReq object	容灾配置信息。

表 5-461 UpdateDisasterRecoveryReq

参数	是否必选	参数类型	描述
dr_sync_period	否	String	容灾同步周期。
send_request	否	Integer	是否发送请求。 最小值：0 最大值：1
primary_to_role	否	String	主集群角色。
reset_action	否	String	设置容灾动作。
standby_to_role	否	String	备集群角色。
dr_status	否	String	容灾状态。

响应参数

状态码： 200

表 5-462 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-463 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

集群容灾同步周期修改为3H，主集群为primary，备集群为standby，容灾动作设为DR_RUNNING：

```
https://{Endpoint}/v2/0536cdee2200d5912f7cc00b877980f1/disaster-recovery/b303b78d-438c-41dd-aaf6-c98eea1fc2a6

{
  "disaster_recovery": {
    "dr_sync_period": "3H",
    "send_request": 0,
    "primary_to_role": "primary",
    "standby_to_role": "standby",
    "reset_action": "DR_RUNNING",
    "dr_status": "running"
  }
}
```

响应示例

状态码： 200

更新容灾配置成功

```
{
  "disaster_recovery": {
    "id": "xxxx"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateDisasterInfoSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
```

```

        .withRegion(DwsRegion.valueOf("cn-north-4"))
        .build();
UpdateDisasterInfoRequest request = new UpdateDisasterInfoRequest();
UpdateDisasterRecoveryRequest body = new UpdateDisasterRecoveryRequest();
UpdateDisasterRecoveryReq disasterRecoverybody = new UpdateDisasterRecoveryReq();
disasterRecoverybody.withDrSyncPeriod("3H")
    .withSendRequest(0)
    .withPrimaryToRole("primary")
    .withResetAction("DR_RUNNING")
    .withStandbyToRole("standby")
    .withDrStatus("running");
body.withDisasterRecovery(disasterRecoverybody);
request.withBody(body);
try {
    UpdateDisasterInfoResponse response = client.updateDisasterInfo(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateDisasterInfoRequest()
        disasterRecoverybody = UpdateDisasterRecoveryReq(
            dr_sync_period="3H",
            send_request=0,
            primary_to_role="primary",
            reset_action="DR_RUNNING",
            standby_to_role="standby",
            dr_status="running"
        )
        request.body = UpdateDisasterRecoveryRequest(
            disaster_recovery=disasterRecoverybody
        )
        response = client.update_disaster_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDisasterInfoRequest{
        drSyncPeriodDisasterRecovery:= "3H"
        sendRequestDisasterRecovery:= int32(0)
        primaryToRoleDisasterRecovery:= "primary"
        resetActionDisasterRecovery:= "DR_RUNNING"
        standbyToRoleDisasterRecovery:= "standby"
        drStatusDisasterRecovery:= "running"
        disasterRecoverybody := &model.UpdateDisasterRecoveryReq{
            DrSyncPeriod: &drSyncPeriodDisasterRecovery,
            SendRequest: &sendRequestDisasterRecovery,
            PrimaryToRole: &primaryToRoleDisasterRecovery,
            ResetAction: &resetActionDisasterRecovery,
            StandbyToRole: &standbyToRoleDisasterRecovery,
            DrStatus: &drStatusDisasterRecovery,
        }
    }
    request.Body = &model.UpdateDisasterRecoveryRequest{
        DisasterRecovery: disasterRecoverybody,
    }
    response, err := client.UpdateDisasterInfo(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	更新容灾配置成功
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.8 灾备切换

功能介绍

该接口用于容灾进行灾备切换操作。“灾备切换”按钮用于在容灾正常情况下主备切换操作。容灾状态为“运行中”时可以执行灾备切换操作。灾备切换需要一定时间，在此期间，原生产集群将不可用。不同场景下进行灾备切换，RPO（Recovery Point Object，灾难发生后，系统和数据必须恢复到的时间点要求。）说明如下：生产集群在“可用”的状态下，RPO=0。生产集群在“不可用”的状态下，无法保证RPO=0，但数据至少可恢复到生产集群“最近容灾成功时间”。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/disaster-recovery/{disaster_recovery_id}/switchover
```

表 5-464 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-465 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-466 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/e59d6b86-9072-46eb-a996-13f8b44994c1/switchover
```

响应示例

状态码： 200

灾备切换下发成功。

```
{
  "disaster_recovery" : {
    "id" : "e59d6b86-9072-46eb-a996-13f8b44994c1"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;
```

```
public class SwitchoverDisasterRecoverySolution {
    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        SwitchoverDisasterRecoveryRequest request = new SwitchoverDisasterRecoveryRequest();
        try {
            SwitchoverDisasterRecoveryResponse response = client.switchoverDisasterRecovery(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = SwitchoverDisasterRecoveryRequest()
        response = client.switchover_disaster_recovery(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchoverDisasterRecoveryRequest{}
    response, err := client.SwitchoverDisasterRecovery(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	灾备切换下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。

状态码	描述
503	服务不可用。

5.12.9 容灾异常切换

功能介绍

该接口用于容灾异常场景下进行主备集群切换操作。“异常切换”按钮用于容灾异常或者生产集群故障情况下主备切换操作。容灾异常切换仅8.1.2及以上集群版本支持。异常切换会将灾备集群升为主，若原生产集群故障后存在部分数据未同步到灾备集群，那灾备集群升主后将缺少这些数据，切换时请确认容灾最后同步时间，谨慎操作。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/disaster-recovery/{disaster_recovery_id}/failover

表 5-467 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-468 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-469 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/e59d6b86-9072-46eb-a996-13f8b44994c1/failover
```

响应示例

状态码： 200

异常切换下发成功。

```
{
  "disaster_recovery" : {
    "id" : "e59d6b86-9072-46eb-a996-13f8b44994c1"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class SwitchFailoverDisasterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        SwitchFailoverDisasterRequest request = new SwitchFailoverDisasterRequest();
        try {
            SwitchFailoverDisasterResponse response = client.switchFailoverDisaster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = SwitchFailoverDisasterRequest()
        response = client.switch_failover_disaster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.SwitchFailoverDisasterRequest{}
response, err := client.SwitchFailoverDisaster(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	异常切换下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.10 恢复容灾

功能介绍

该接口用于主备集群进行异常切换，备集群恢复可用状态后进行的容灾恢复操作。容灾恢复仅8.1.2及以上集群版本支持。容灾恢复会删除灾备集群数据与新生产集群重新建立容灾关系。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/disaster-recovery/{disaster_recovery_id}/recovery

表 5-470 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-471 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-472 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/e59d6b86-9072-46eb-a996-13f8b44994c1/recovery
```

响应示例

状态码： 200

恢复容灾下发成功。

```
{
  "disaster_recovery": {
    "id": "e59d6b86-9072-46eb-a996-13f8b44994c1"
  }
}
```


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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class RestoreDisasterSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        RestoreDisasterRequest request = new RestoreDisasterRequest();
        try {
            RestoreDisasterResponse response = client.restoreDisaster(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = RestoreDisasterRequest()
    response = client.restore_disaster(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestoreDisasterRequest{}
    response, err := client.RestoreDisaster(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	恢复容灾下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.11 停止容灾

功能介绍

该接口用于停止容灾操作。容灾状态为“运行中”和“停止失败”时可以执行停止容灾操作。停止后，将无法进行数据同步，请谨慎操作。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v2/{project_id}/disaster-recovery/{disaster_recovery_id}/pause
```

表 5-473 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

状态码： 200

表 5-474 响应 Body 参数

参数	参数类型	描述
disaster_recovery	DisasterRecoveryId object	容灾ID对象。

表 5-475 DisasterRecoveryId

参数	参数类型	描述
id	String	容灾ID。

请求示例

```
POST https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/e59d6b86-9072-46eb-a996-13f8b44994c1/pause
```

响应示例

状态码： 200

停止容灾下发成功。

```
{
  "disaster_recovery": {
    "id": "e59d6b86-9072-46eb-a996-13f8b44994c1"
  }
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class PauseDisasterRecoverySolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
PauseDisasterRecoveryRequest request = new PauseDisasterRecoveryRequest();
try {
    PauseDisasterRecoveryResponse response = client.pauseDisasterRecovery(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = PauseDisasterRecoveryRequest()
        response = client.pause_disaster_recovery(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"

```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.PauseDisasterRecoveryRequest{}
    response, err := client.PauseDisasterRecovery(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	停止容灾下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.12 检查容灾名称

功能介绍

该接口用于查询容灾名称是否可用。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v2/{project_id}/disaster-recovery/check-name
```

表 5-476 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 5-477 Query 参数

参数	是否必选	参数类型	描述
dr_name	是	String	容灾名称。
type	否	String	容灾类型。
standby_region	否	String	备集群所在Region。
standby_project_id	否	String	备集群所在项目ID。

请求参数

无

响应参数

无

请求示例

```
https://{Endpoint}/rds/v2/0536cdee2200d5912f7cc00b877980f1/disaster-recovery/check-name?  
dr_name=dws-it
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class CheckDisasterNameSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        CheckDisasterNameRequest request = new CheckDisasterNameRequest();
        request.withDrName("<dr_name>");
        request.withType("<type>");
        request.withStandbyRegion("<standby_region>");
        request.withStandbyProjectId("<standby_project_id>");
        try {
            CheckDisasterNameResponse response = client.checkDisasterName(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```



```
credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = CheckDisasterNameRequest()
    request.dr_name = "<dr_name>"
    request.type = "<type>"
    request.standby_region = "<standby_region>"
    request.standby_project_id = "<standby_project_id>"
    response = client.check_disaster_name(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.CheckDisasterNameRequest{}
    request.DrName = "<dr_name>"
    typeRequest := "<type>"
    request.Type = &typeRequest
    standbyRegionRequest := "<standby_region>"
    request.StandbyRegion = &standbyRegionRequest
    standbyProjectIdRequest := "<standby_project_id>"
    request.StandbyProjectId = &standbyProjectIdRequest
    response, err := client.CheckDisasterName(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	容灾名称检查成功。
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.12.13 删除容灾

功能介绍

该接口用于删除容灾操作。容灾状态为“创建失败”、“未启动”、“启动失败”、“已停止”、“停止失败”和“异常”时可以执行删除容灾操作。删除后，将无法进行数据同步，且不可恢复，请谨慎操作。仅支持DWS 2.0集群。

调用方法

请参见[如何调用API](#)。

URI

```
DELETE /v2/{project_id}/disaster-recovery/{disaster_recovery_id}
```

表 5-478 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 。
disaster_recovery_id	是	String	容灾ID。

请求参数

无

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/disaster-recovery/b5c45780-1006-49e3-b2d5-b3229975bbc7
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DeleteDisasterRecoverySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDisasterRecoveryRequest request = new DeleteDisasterRecoveryRequest();
        try {
            DeleteDisasterRecoveryResponse response = client.deleteDisasterRecovery(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = DeleteDisasterRecoveryRequest()  
        response = client.delete_disaster_recovery(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"  
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(  
        dws.DwsClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())
```

```
request := &model.DeleteDisasterRecoveryRequest{}
response, err := client.DeleteDisasterRecovery(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	删除容灾下发成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.13 任务管理

5.13.1 查询 job 进度

功能介绍

该接口用于查询job进度信息。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/job/{job_id}
```

表 5-479 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 5-480 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID。
job_name	String	任务名称。
begin_time	String	任务开始时间。
end_time	String	任务结束时间。
status	String	任务当前状态。
failed_code	String	任务失败错误码。
failed_detail	String	任务失败错误详情。
progress	String	任务进度。

请求示例

`https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/job/2c9080e8845b207101845b245e1e0001`

响应示例

状态码： 200

任务进度

```
{
  "status": "FAIL",
  "progress": "9%",
  "job_id": "2c9080e88459fa44018459fbeb600001",
  "job_name": "ecfClusterElbCreateJob",
  "begin_time": "2022-11-09T20:25:00",
  "end_time": "2022-11-09T20:30:00",
  "failed_code": "CreateELBTask-fail:DWS.0114",
```

```
"failed_detail" : "DWS.0114:ELB private IP is not configured."  
}
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListJobDetailsSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListJobDetailsRequest request = new ListJobDetailsRequest();  
        try {  
            ListJobDetailsResponse response = client.listJobDetails(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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListJobDetailsRequest()
        response = client.list_job_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListJobDetailsRequest{}
    response, err := client.ListJobDetails(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	查询任务进度成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14 主机监控

5.14.1 查询主机概览

功能介绍

该接口用于查询主机监控概览信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/dms/host-overview

表 5-481 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 5-482 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群ID。获取方法，请参见 获取集群ID 。
instance_name	否	String	实例名称。
limit	是	Integer	数据条目数。取值范围为1~2147483647。
offset	是	Integer	数据偏移量。取值范围为0~2147483646。

请求参数

无

响应参数

表 5-483 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of HostOverviewResponse objects	主机概览列表。

表 5-484 HostOverviewResponse

参数	参数类型	描述
instance_name	String	实例名称
host_name	String	主机名称
host_stat	String	主机状态，其中包括： <ul style="list-style-type: none"> ● ONLINE：在线 ● OFFLINE：离线
work_ip	String	IP地址。
mem_free	double	系统中未使用的内存(GB)。
mem_total	double	总内存(GB)。
mem_usage	double	内存使用率(GB)。
mem_cached	double	缓存内存(GB)。

参数	参数类型	描述
mem_buffer	double	缓冲内存(MB)。
swap_free	double	ram暂存在swap中的大小(GB)。
swap_total	double	交换空间总和(GB)。
cpu_usage	double	CPU使用率(%)。
cpu_usage_sys	double	系统CPU占用率(%)。
cpu_usage_usr	double	用户CPU占用率(%)。
cpu_idle	double	空闲CPU占用率(%)。
cpu_iowait	double	IO等待(%)。
disk_usage_avg	double	磁盘平均使用率(%)。
disk_total	double	磁盘总容量(GB)。
disk_used	double	磁盘使用容量(GB)。
disk_available	double	磁盘可用容量(GB)。
disk_io	double	磁盘IO(KB/s)。
disk_io_read	double	磁盘读速率(KB/s)。
disk_io_write	double	磁盘写速率(KB/s)。
tcp_resend_rate	double	TCP协议栈重传率(%)。
net_io	double	网络IO(KB/s)。

请求示例

<https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/host-overview?limit=1&offset=0>

响应示例

状态码：200

查询主机概览成功。

```
[
  {
    "instance_name": "d30035376-dms-test07-dws-cn-cn-1-1",
    "host_name": "host-172-16-11-62",
    "host_stat": "ONLINE",
    "work_ip": "172.16.106.168",
    "mem_free": 1.45,
    "mem_total": 14.89,
    "mem_usage": 26.4,
    "mem_cached": 9.26,
    "mem_buffer": 249.34,
    "swap_free": 0,
    "swap_total": 0,
    "cpu_usage": 8.93,
```

```
"cpu_usage_sys": 3.37,  
"cpu_usage_usr": 5.56,  
"cpu_idle": 90.57,  
"cpu_iowait": 0.12,  
"disk_usage_avg": 4.44,  
"disk_total": 548.76,  
"disk_used": 24.38,  
"disk_available": 522,  
"disk_io": 19.75,  
"disk_io_read": 0,  
"disk_io_write": 19.75,  
"tcp_resend_rate": 0,  
"net_io": 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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListHostOverviewSolution {  
  
    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);  
  
        DwsClient client = DwsClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListHostOverviewRequest request = new ListHostOverviewRequest();  
        request.withClusterId("<cluster_id>");  
        request.withInstanceName("<instance_name>");  
        request.withLimit(<limit>);  
        request.withOffset(<offset>);  
        try {  
            ListHostOverviewResponse response = client.listHostOverview(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListHostOverviewRequest()
        request.cluster_id = "<cluster_id>"
        request.instance_name = "<instance_name>"
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_host_overview(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListHostOverviewRequest{}
clusterIdRequest:= "<cluster_id>"
request.ClusterId = &clusterIdRequest
instanceNameRequest:= "<instance_name>"
request.InstanceName = &instanceNameRequest
request.Limit = int32(<limit>)
request.Offset = int32(<offset>)
response, err := client.ListHostOverview(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	查询主机概览成功。
204	处理成功，但没有返回任何内容。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.2 查询磁盘信息

功能介绍

该接口用于查询主机监控磁盘信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/dms/disk

表 5-485 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 5-486 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群ID。获取方法，请参见 获取集群ID 。
instance_name	否	String	实例名称。
instance_id	否	String	实例ID。
limit	是	Integer	数据条目数。取值范围为1~2147483647。
offset	是	Integer	数据偏移量。取值范围为0~2147483646。

请求参数

无

响应参数

表 5-487 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of DiskResp objects	主机磁盘列表。

表 5-488 DiskResp

参数	参数类型	描述
instance_name	String	实例名称。
instance_id	String	实例ID

参数	参数类型	描述
host_name	String	主机名称。
disk_name	String	磁盘名称。
disk_type	String	磁盘类型(系统盘、数据盘、日志盘)。
total	double	磁盘总容量(GB)。
used	double	磁盘已使用容量(GB)。
available	double	磁盘可用容量(GB)。
used_percentage	double	磁盘使用率(%)。
await	double	IO等待时间(ms)。
svctm	double	IO服务时间(ms)。
util	double	IO使用率(%)。
read_rate	double	磁盘读速率(KB/s)。
write_rate	double	磁盘写速率(KB/s)。

请求示例

```
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/disk?limit=1&offset=0
```

响应示例

```
[
  {
    "instance_name": "d30035376-dms-test07-dws-cn-cn-1-1",
    "instance_id": "1",
    "host_name": "host-172-16-11-62",
    "disk_name": "vda",
    "disk_type": "system",
    "total": 49,
    "used": 5.64,
    "available": 40.99,
    "used_percentage": 11.51,
    "await": 0.21,
    "svctm": 0.03,
    "util": 0.01,
    "read_rate": 0,
    "write_rate": 52.72
  }
]
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListHostDiskSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListHostDiskRequest request = new ListHostDiskRequest();
        request.withClusterId("<cluster_id>");
        request.withInstanceId("<instance_id>");
        request.withInstanceName("<instance_name>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListHostDiskResponse response = client.listHostDisk(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")
```

```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListHostDiskRequest()
    request.cluster_id = "<cluster_id>"
    request.instance_id = "<instance_id>"
    request.instance_name = "<instance_name>"
    request.limit = <limit>
    request.offset = <offset>
    response = client.list_host_disk(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListHostDiskRequest{}
    clusterIdRequest := "<cluster_id>"
    request.ClusterId = &clusterIdRequest
    instanceIdRequest := "<instance_id>"
    request.InstanceId = &instanceIdRequest
    instanceNameRequest := "<instance_name>"
    request.InstanceName = &instanceNameRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    response, err := client.ListHostDisk(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	查询磁盘信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.3 获取网卡状态

功能介绍

该接口用于获取主机监控网卡状态。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1.0/{project_id}/dms/net
```

表 5-489 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 5-490 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	集群ID。获取方法，请参见 获取集群ID 。
instance_name	否	String	实例名称。
limit	是	Integer	数据条目数。取值范围为1~2147483647。
offset	是	Integer	数据偏移量。取值范围为0~2147483646。

请求参数

无

响应参数

表 5-491 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of NetResp objects	主机网卡列表。

表 5-492 NetResp

参数	参数类型	描述
virtual_cluster_id	Integer	虚拟集群ID。
ctime	Long	查询时间。
host_id	Integer	主机ID。
host_name	String	主机名称。
instance_name	String	实例名称。
interface_name	String	网卡名称。
up	Boolean	网卡状态（true代表up/false代表down）。
speed	Long	网卡速度(Mbps)。
recv_packets	Long	接收包数(个)。
send_packets	Long	发送包数(个)。

参数	参数类型	描述
recv_drop	Long	接收丢包数(个)。
recv_rate	Double	接收速率(KB/s)。
send_rate	Double	发送速率(KB/s)。
io_rate	Double	网络速率(KB/s)。

请求示例

```
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/net?limit=1&offset=0
```

响应示例

```
[
  {
    "virtual_cluster_id": 23,
    "ctime": 1668071223000,
    "host_id": 34,
    "host_name": "host-172-16-11-62",
    "instance_name": "d30035376-dms-test07-dws-cn-cn-1-1",
    "interface_name": "eth2",
    "up": true,
    "speed": -1,
    "recv_packets": 490595835,
    "send_packets": 533376765,
    "recv_drop": 0,
    "recv_rate": 16.35,
    "send_rate": 11.71,
    "io_rate": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListHostNetSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListHostNetRequest request = new ListHostNetRequest();
request.withClusterId("<cluster_id>");
request.withInstanceName("<instance_name>");
request.withLimit(<limit>);
request.withOffset(<offset>);
try {
    ListHostNetResponse response = client.listHostNet(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListHostNetRequest()
        request.cluster_id = "<cluster_id>"
        request.instance_name = "<instance_name>"
        request.limit = <limit>
        request.offset = <offset>
        response = client.list_host_net(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListHostNetRequest{}
    clusterIdRequest := "<cluster_id>"
    request.ClusterId = &clusterIdRequest
    instanceNameRequest := "<instance_name>"
    request.InstanceName = &instanceNameRequest
    request.Limit = int32(<limit>)
    request.Offset = int32(<offset>)
    response, err := client.ListHostNet(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	获取网卡状态成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。

状态码	描述
500	服务内部错误。
503	服务不可用。

5.14.4 查询性能监控指标

功能介绍

该接口用于获取监控指标。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/dms/metric-data/indicators

表 5-493 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

无

响应参数

状态码： 200

表 5-494 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of IndicatorInfo objects	监控指标列表。

表 5-495 IndicatorInfo

参数	参数类型	描述
indicator_name	String	监控指标名称。
plugin_name	String	采集模块名称。
default_collect_rate	String	默认采集频率。
support_datastore_version	String	支持的集群版本。

请求示例

```
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/metric-data/indicators
```

响应示例

状态码： 200

openapi获取监控指标成功

```
[ {
  "indicator_name": "cluster.cpu_usage",
  "plugin_name": "CpuStat",
  "default_collect_rate": "30",
  "support_datastore_version": "8.0.0"
}, {
  "indicator_name": "cluster.sql_probe_single",
  "plugin_name": "SqlProbeTask",
  "default_collect_rate": "30",
  "support_datastore_version": "8.0.0"
} ]
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListMonitorIndicatorsSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListMonitorIndicatorsRequest request = new ListMonitorIndicatorsRequest();
try {
    ListMonitorIndicatorsResponse response = client.listMonitorIndicators(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListMonitorIndicatorsRequest()
        response = client.list_monitor_indicators(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListMonitorIndicatorsRequest{}
    response, err := client.ListMonitorIndicators(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	获取监控指标成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.5 查询历史监控数据

功能介绍

该接口用于查询历史监控数据。

调用方法

请参见[如何调用API](#)。

URI

GET /v1.0/{project_id}/dms/metric-data

表 5-496 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 5-497 Query 参数

参数	是否必选	参数类型	描述
from	是	String	开始时间。
to	是	String	结束时间。
function	否	String	取值方法。
period	否	String	取值周期。
indicator_name	是	String	监控指标名称。
dim0	是	String	第一层级。
dim1	否	String	第二层级。

请求参数

无

响应参数

状态码： 200

表 5-498 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of TrendQueryDataResponseBody objects	历史监控数据列表。

表 5-499 TrendQueryDataResponseBody

参数	参数类型	描述
query_time	Long	查询时间。
indicator_name	String	监控指标名称。查询性能监控指标接口响应参数中“indicator_name”的值。
object_id	String	监控对象ID。根据indicator_name传入不同参数，详见请求示例。
unit	String	单位。
sub_object_id	String	次级监控ID。根据indicator_name传入不同参数，详见请求示例。
data_points	Array of TrendQueryData objects	节点数据。

表 5-500 TrendQueryData

参数	参数类型	描述
result	String	查询结果。
timestamp	Long	时间戳。

请求示例

indicator_name存在以下四种类型，根据不同类型的指标传参不同，示例如下：

- cluster.*:**
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/metric-data?from=1671058114000&to=1671058144000&indicator_name=cluster.cpu_usage&dim0=cluster_id,fe658e95-5284-4adf-b6aa-7729b563dxxx
- database.*:**
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/metric-data?from=1671058114000&to=1671058144000&indicator_name=database.active_sessions&dim0=cluster_id,fe658e95-5284-4adf-b6aa-7729b563dxxx&dim1=db_name,progres
- host.*:**

```
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/metric-data?  
from=1671058114000&to=1671058144000&indicator_name=host.host_disk_await&dim0=instance_name,dws-data-dws-cn-cn-3-1&dim1=disk_type,system
```

- **node.*:**

```
https://{Endpoint}/v1.0/05f2cff45100d5112f4bc00b794ea08e/dms/metric-data?  
from=1671058114000&to=1671058144000&indicator_name=node.dynamic_mem_usage&dim0=cluster_id,fe658e95-5284-4adf-b6aa-7729b563dxxx&dim1=node_name,cn_5003
```

响应示例

状态码： 200

```
[ {  
  "query_time" : 1671160536504,  
  "indicator_name" : "cluster cpu_usage",  
  "object_id" : "fe658e95-5284-4adf-b6aa-7729b563dcb7",  
  "unit" : "percentage",  
  "data_points" : [ {  
    "result" : "10.20",  
    "timestamp" : 1671058114000  
  }, {  
    "result" : "10.20",  
    "timestamp" : 1671058144000  
  } ]  
}, {  
  "query_time" : 1671160536504,  
  "indicator_name" : "cluster cpu_mem",  
  "object_id" : "fe658e95-5284-4adf-b6aa-7729b563dcb7",  
  "unit" : "percentage",  
  "data_points" : [ {  
    "result" : "10.20",  
    "timestamp" : 1671058114000  
  }, {  
    "result" : "10.20",  
    "timestamp" : 1671058144000  
  } ]  
} ]
```

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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ListMonitorIndicatorDataSolution {  
  
  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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListMonitorIndicatorDataRequest request = new ListMonitorIndicatorDataRequest();
request.withFrom("<from>");
request.withTo("<to>");
request.withFunction("<function>");
request.withPeriod("<period>");
request.withIndicatorName("<indicator_name>");
request.withDim0("<dim0>");
request.withDim1("<dim1>");
try {
    ListMonitorIndicatorDataResponse response = client.listMonitorIndicatorData(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListMonitorIndicatorDataRequest()
        request._from = "<from>"
        request.to = "<to>"
        request.function = "<function>"
        request.period = "<period>"
        request.indicator_name = "<indicator_name>"
        request.dim0 = "<dim0>"
        request.dim1 = "<dim1>"
        response = client.list_monitor_indicator_data(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListMonitorIndicatorDataRequest{}
    request.From = "<from>"
    request.To = "<to>"
    functionRequest := "<function>"
    request.Function = &functionRequest
    periodRequest := "<period>"
    request.Period = &periodRequest
    request.IndicatorName = "<indicator_name>"
    request.Dim0 = "<dim0>"
    dim1Request := "<dim1>"
    request.Dim1 = &dim1Request
    response, err := client.ListMonitorIndicatorData(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	查询历史监控数据成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.6 查询表倾斜或脏页率信息

功能介绍

该接口用于查询表倾斜或脏页率信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/dms/tables/statistic

表 5-501 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-502 Query 参数

参数	是否必选	参数类型	描述
rate_type	是	String	查询类型。固定取值。 <ul style="list-style-type: none"> skew：表倾斜率。 dirtyPage：表脏页率。

参数	是否必选	参数类型	描述
offset	是	Integer	偏移量，表示从此偏移量开始查询，offset>=0。
limit	是	Integer	每页显示的条目数量。
order_by	否	String	排序字段。固定取值。 <ul style="list-style-type: none"> table_size: 表大小。 rate: 表倾斜率或脏页率。
sort_by	否	String	正序还是倒叙。固定取值。 <ul style="list-style-type: none"> ASC: 正序。 DESC: 倒序。
filter	否	String	查询条件。固定取值。 <ul style="list-style-type: none"> db_name: 数据库名称。 schema_name: schema名称。 table_name: 表名。 table_owner: 所属用户。
value	否	String	查询条件的取值。

请求参数

无

响应参数

状态码： 200

表 5-503 响应 Body 参数

参数	参数类型	描述
collect_time	Long	数据采集时间毫秒级时间戳。
data	Array of ListTablesStatisticDto objects	表倾斜率或脏页率列表。
count	Integer	总列表大小。

表 5-504 ListTablesStatisticDto

参数	参数类型	描述
db_name	String	数据库名称。
schema_name	String	schema名称。
table_name	String	表名。
table_owner	String	所属用户。
table_size	String	表大小。
skew_rate	Double	表倾斜率。
dirty_page_rate	Double	脏页率。

请求示例

```
https://{Endpoint}/v1/{project_id}/clusters/{cluster_id}/dms/tables/statistic?rate_type=dirtyPage&filter=&value=&offset=0&limit=1&order_by=&sort_by=ASC
```

响应示例

状态码： 200

```
{
  "collect_time" : 0,
  "data" : [ {
    "db_name" : "postgres",
    "schema_name" : "pmk",
    "table_name" : "pmk_snapshot_coordinator_stat",
    "table_owner" : "Ruby",
    "table_size" : "224 KB",
    "skew_rate" : null,
    "dirty_page_rate" : 0.77
  } ],
  "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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListTablesStatisticSolution {
```

```
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);

    DwsClient client = DwsClient.newBuilder()
        .withCredential(auth)
        .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
        .build();
    ListTablesStatisticRequest request = new ListTablesStatisticRequest();
    request.withRateType("<rate_type>");
    request.withOffset("<offset>");
    request.withLimit("<limit>");
    request.withOrderBy("<order_by>");
    request.withSortBy("<sort_by>");
    request.withFilter("<filter>");
    request.withValue("<value>");
    try {
        ListTablesStatisticResponse response = client.listTablesStatistic(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
```

```

request = ListTablesStatisticRequest()
request.rate_type = "<rate_type>"
request.offset = <offset>
request.limit = <limit>
request.order_by = "<order_by>"
request.sort_by = "<sort_by>"
request.filter = "<filter>"
request.value = "<value>"
response = client.list_tables_statistic(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTablesStatisticRequest{}
    request.RateType = "<rate_type>"
    request.Offset = int32(<offset>)
    request.Limit = int32(<limit>)
    orderByRequest := "<order_by>"
    request.OrderBy = &orderByRequest
    sortByRequest := "<sort_by>"
    request.SortBy = &sortByRequest
    filterRequest := "<filter>"
    request.Filter = &filterRequest
    valueRequest := "<value>"
    request.Value = &valueRequest
    response, err := client.ListTablesStatistic(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	查询表倾斜率或脏页率成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.7 查询 SQL 列表

功能介绍

该接口用于查询实时SQL列表。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/clusters/{cluster_id}/dms/queries

表 5-505 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-506 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_id	是	String	集群ID。
project_id	是	String	项目ID。
offset	是	String	偏移量，表示从此偏移量开始查询，offset>=0。
limit	是	String	每页显示的条目数量。
conditions	是	Array of ListQueriesCondition objects	查询条件数组。
order_by	否	String	排序字段。
target	是	String	固定值db_queries。

表 5-507 ListQueriesCondition

参数	是否必选	参数类型	描述
field	是	String	字段名称。
value	是	String	字段值。
operator	是	String	比较方式： <ul style="list-style-type: none"> • String类型参数： <ul style="list-style-type: none"> - = - != - like - not like • int类型参数： <ul style="list-style-type: none"> - = - != - > - < - >= - <= • boolean类型参数： <ul style="list-style-type: none"> - = - !=

响应参数

状态码： 200

表 5-508 响应 Body 参数

参数	参数类型	描述
code	Integer	响应码。
msg	String	响应信息。
data	ListQueriesData object	响应数据。
count	Integer	总条数。

表 5-509 ListQueriesData

参数	参数类型	描述
queries	Array of ListQueriesData objects	查询数据列表。
status	ListQueriesStatus object	统计数据。

表 5-510 ListQueriesDto

参数	参数类型	描述
virtual_cluster_id	Integer	虚拟集群ID。
ctime	Long	采集时间。
pid	String	会话id。
inst_name	String	实例名称。
waiting	Boolean	如果后台当前正等待锁则为true。
enqueue	String	工作负载管理资源状态。
warning	String	主要显示如下几类告警信息以及sql自诊断调优相关告警。
query	String	查询语句。
lane	String	快慢车道 (fast or slow)。
db_name	String	数据库名称。

参数	参数类型	描述
priority	String	job在资源池中的优先级，取值：1,2,4,8（rush、high、medium、low）。
query_id	String	语句执行使用的内部query_id。
query_band	String	用于标示作业类型，可通过guc参数query_band进行设置，默认为空字符串。
job_name	String	这个值是从query_band的字段中取出来的，位置0。
job_inst	String	这个值是从query_band的字段中取出来的，位置1。
user_name	String	连接到后端的用户名。
application_name	String	连接到后端的应用名。
client_address	String	连接到后端的客户端的ip地址。
client_hostname	String	客户端的主机名。
client_port	String	客户端用于与后端通讯的tcp端口号。
start_time	Long	语句执行的开始时间。
block_time	Long	语句执行前的阻塞时间（单位ms）。
duration	Long	语句已经执行的时间（单位ms）。
estimate_total_time	Long	语句执行预估总时间（单位ms）。
estimate_left_time	Long	语句执行预估剩余时间（单位ms）。
resource_pool	String	用户使用的资源池。
control_group	String	语句所使用的cgroup。
min_peak_memory	Integer	语句在所有dn上的最小内存峰值（单位mb）。
max_peak_memory	Integer	语句在所有dn上的最大内存峰值（单位mb）。
average_peak_memory	Integer	语句执行过程中的内存使用平均值（单位mb）。
memory_skew_percent	Integer	语句在各dn间的内存使用倾斜率。
estimate_memory	Integer	语句预估使用内存（单位mb）。

参数	参数类型	描述
spill_info	String	语句在所有dn上的下盘信息。
min_spill_size	Integer	若发生下盘，所有dn上下盘的最小数据量 (单位mb) 默认为0。
max_spill_size	Integer	若发生下盘，所有dn上下盘的最大数据量 (单位mb) 默认为0。
average_spill_size	Integer	若发生下盘，所有dn上下盘的平均数据量 (单位mb) 默认为0。
spill_skew_percent	Integer	若发生下盘，dn间下盘倾斜率。
min_dn_time	Long	语句在所有dn上的最小执行时间 (单位ms)。
max_dn_time	Long	语句在所有dn上的最大执行时间 (单位ms)。
average_dn_time	Long	语句在所有dn上的平均执行时间 (单位ms)。
dn_time_skew_percent	Integer	语句在各dn间的执行时间倾斜率。
min_cpu_time	Long	语句在所有dn上的最小cpu时间 (单位ms)。
max_cpu_time	Long	语句在所有dn上的最大cpu时间 (单位ms)。
total_cpu_time	Long	语句在所有dn上的cpu总时间 (单位ms)。
cpu_skew_percent	Integer	语句在各dn间的cpu时间倾斜率。
average_peak_iops	Integer	语句在所有dn上的每秒平均io峰值 (列存单位是次/s, 行存单位是万次/s)。
iops_skew_percent	Integer	语句在dn间的io倾斜率。
max_peak_iops	Integer	语句在所有dn上的每秒最大io峰值 (列存单位是次/s, 行存单位是万次/s)。
min_peak_iops	Integer	语句在所有dn上的每秒最小io峰值 (列存单位是次/s, 行存单位是万次/s)。
query_plan	String	查询计划。
query_status	String	当前查询语句的实时运行状态 (active, idle, idle in transaction, idle in transaction(aborted), fastpath function call, disabled)。
wlm_status	String	当前查询语句在资源池上的运行状态 (pending, running, finished, aborted, active, unknown)。
wlm_attr	String	语句的属性 (ordinary, simple, complicated, internal)

参数	参数类型	描述
system_query	Boolean	是否系统查询。
backend_start	Long	该过程开始的时间，即当客户端连接服务器时。
elapsed_time	Long	到目前为止的执行时间。
curr_xact_start	Long	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
state_change	Long	上次状态改变的时间。
query_start	Long	语句执行的开始时间。
query_elapsed_time	Long	语句当前为止的实际执行时间(单位：s)。

表 5-511 ListQueriesStatus

参数	参数类型	描述
average_query_waiting_time	Double	平均查询等待时间。
average_time_consumption_of_queries	Double	平均查询耗时。
average_time_consumption_of_sessions	Double	平均会话耗时。
queries_count	Long	查询数量。
session_count	Long	会话数量。

请求示例

https://{Endpoint}/v2/{project_id}/clusters/{cluster_id}/dms/queries

```
{
  "offset" : 0,
  "limit" : 1,
  "cluster_id" : "cluster_id",
  "project_id" : "project_id",
  "conditions" : [ {
    "field" : "userName",
    "value" : "用户名称",
    "operator" : "="
  }, {
    "field" : "applicationName",
    "value" : "应用名称",
    "operator" : "<>"
  }, {

```

```
"field" : "dbName",
"value" : "数据库名称",
"operator" : "="
}, {
"field" : "resourcePool",
"value" : "资源池",
"operator" : "<>"
}, {
"field" : "queryStatus",
"value" : "查询状态",
"operator" : "="
}, {
"field" : "enqueue",
"value" : "排队状态",
"operator" : "<>"
}, {
"field" : "lane",
"value" : "快慢车道",
"operator" : "="
}, {
"field" : "instName",
"value" : "接入CN",
"operator" : "<>"
}, {
"field" : "pid",
"value" : "会话ID",
"operator" : "="
}, {
"field" : "blockTime",
"value" : "1",
"operator" : "="
}, {
"field" : "duration",
"value" : "2",
"operator" : "<>"
}, {
"field" : "minCpuTime",
"value" : "3",
"operator" : ">"
}, {
"field" : "maxCpuTime",
"value" : "4",
"operator" : "<"
}, {
"field" : "totalCpuTime",
"value" : "5",
"operator" : ">="
}, {
"field" : "cpuSkewPercent",
"value" : "6",
"operator" : "<="
}, {
"field" : "spillInfo",
"value" : "dn下盘信息",
"operator" : "="
}, {
"field" : "minSpillSize",
"value" : "7",
"operator" : "<>"
}, {
"field" : "maxSpillSize",
"value" : "8",
"operator" : ">"
}, {
"field" : "averageSpillSize",
"value" : "9",
"operator" : "<"
}, {
"field" : "spillSkewPercent",
```

```
"value": "10",
"operator": ">="
}, {
  "field": "queryBand",
  "value": "作业类型",
  "operator": "<>"
}, {
  "field": "jobName",
  "value": "任务名称",
  "operator": "="
}, {
  "field": "jobInst",
  "value": "任务实例",
  "operator": "<>"
}, {
  "field": "clientHostname",
  "value": "主机名称",
  "operator": "="
}, {
  "field": "clientPort",
  "value": "TCP端口",
  "operator": "<>"
}, {
  "field": "waiting",
  "value": "是否等待",
  "operator": "="
}, {
  "field": "estimateTotalTime",
  "value": "11",
  "operator": "="
}, {
  "field": "estimateLeftTime",
  "value": "12",
  "operator": "<>"
}, {
  "field": "controlGroup",
  "value": "cgroup",
  "operator": "like"
}, {
  "field": "minPeakMemory",
  "value": "13",
  "operator": "="
}, {
  "field": "maxPeakMemory",
  "value": "14",
  "operator": "<>"
}, {
  "field": "averagePeakMemory",
  "value": "15",
  "operator": ">"
}, {
  "field": "memorySkewPercent",
  "value": "16",
  "operator": "<"
}, {
  "field": "estimateMemory",
  "value": "17",
  "operator": ">="
}, {
  "field": "minDnTime",
  "value": "18",
  "operator": "<="
}, {
  "field": "maxDnTime",
  "value": "19",
  "operator": "="
}, {
  "field": "averageDnTime",
  "value": "20",
```

```
"operator": "<>"
}, {
  "field": "dntimeSkewPercent",
  "value": "21",
  "operator": ">"
}, {
  "field": "warning",
  "value": "告警",
  "operator": "="
}, {
  "field": "averagePeaklops",
  "value": "22",
  "operator": "<>"
}, {
  "field": "iopsSkewPercent",
  "value": "23",
  "operator": ">"
}, {
  "field": "wlmStatus",
  "value": "语句运行状态",
  "operator": "="
}, {
  "field": "wlmAttrib",
  "value": "语句属性",
  "operator": "not like"
}],
"order_by": "duration asc",
"target": "db_queries"
}
```

响应示例

状态码: 200

```
{
  "code": 0,
  "msg": "OK",
  "count": 0,
  "data": {
    "queries": [ {
      "ctime": 1699062726000,
      "pid": "140535026615872",
      "waiting": false,
      "duration": 0,
      "enqueue": "",
      "warning": "",
      "query": "WLM fetch collect info from data nodes",
      "lane": "",
      "priority": null,
      "virtual_cluster_id": 0,
      "inst_name": "cn_5002",
      "db_name": "postgres",
      "query_id": "145522562959854219",
      "query_band": "",
      "job_name": "",
      "job_inst": "",
      "user_name": "Ruby",
      "application_name": "workload",
      "client_address": "",
      "client_hostname": "",
      "client_port": "",
      "start_time": 0,
      "block_time": 0,
      "estimate_total_time": 0,
      "estimate_left_time": 0,
      "resource_pool": "default_pool",
      "control_group": "",
      "min_peak_memory": 0,
      "max_peak_memory": 0,
    }
  ]
}
```

```
"average_peak_memory" : 0,
"memory_skew_percent" : 0,
"estimate_memory" : 0,
"spill_info" : "",
"min_spill_size" : 0,
"max_spill_size" : 0,
"average_spill_size" : 0,
"spill_skew_percent" : 0,
"min_dn_time" : 0,
"max_dn_time" : 0,
"average_dn_time" : 0,
"dntime_skew_percent" : 0,
"min_cpu_time" : 0,
"max_cpu_time" : 0,
"total_cpu_time" : 0,
"cpu_skew_percent" : 0,
"average_peak_iops" : 0,
"iops_skew_percent" : 0,
"max_peak_iops" : 0,
"min_peak_iops" : 0,
"query_plan" : null,
"query_status" : "active",
"wlm_status" : "",
"wlm_attrib" : "",
"system_query" : true,
"backend_start" : 1698998138,
"elapsed_time" : 64585,
"curr_xact_start" : 1699062726,
"state_change" : 1698998142,
"query_start" : 1698998142,
"query_elapsed_time" : -1
}],
"status" : {
  "session_count" : 19,
  "average_time_consumption_of_sessions" : 51297.58,
  "queries_count" : 19,
  "average_time_consumption_of_queries" : 48799.8,
  "average_query_waiting_time" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class ListQueriesSolution {

    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListQueriesRequest request = new ListQueriesRequest();
ListQueriesRequestBody body = new ListQueriesRequestBody();
List<ListQueriesCondition> listbodyConditions = new ArrayList<>();
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("userName")
        .withValue("用户名称")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("applicationName")
        .withValue("应用名称")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("dbName")
        .withValue("数据库名称")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("resourcePool")
        .withValue("资源池")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("queryStatus")
        .withValue("查询状态")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("enqueue")
        .withValue("排队状态")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("lane")
        .withValue("快慢车道")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("instName")
        .withValue("接入CN")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("pid")
        .withValue("会话ID")
        .withOperator("=")
```



```
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("blockTime")
        .withValue("1")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("duration")
        .withValue("2")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("minCpuTime")
        .withValue("3")
        .withOperator(">")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("maxCpuTime")
        .withValue("4")
        .withOperator("<")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("totalCpuTime")
        .withValue("5")
        .withOperator(">=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("cpuSkewPercent")
        .withValue("6")
        .withOperator("<=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("spillInfo")
        .withValue("dn下盘信息")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("minSpillSize")
        .withValue("7")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("maxSpillSize")
        .withValue("8")
        .withOperator(">")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("averageSpillSize")
        .withValue("9")
        .withOperator("<")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("spillSkewPercent")
        .withValue("10")
        .withOperator(">=")
);
listbodyConditions.add(
    new ListQueriesCondition()

```

```

        .withField("queryBand")
        .withValue("作业类型")
        .withOperator("<>")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("jobName")
            .withValue("任务名称")
            .withOperator("=")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("jobInst")
            .withValue("任务实例")
            .withOperator("<>")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("clientHostname")
            .withValue("主机名称")
            .withOperator("=")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("clientPort")
            .withValue("TCP端口")
            .withOperator("<>")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("waiting")
            .withValue("是否等待")
            .withOperator("=")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("estimateTotalTime")
            .withValue("11")
            .withOperator("=")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("estimateLeftTime")
            .withValue("12")
            .withOperator("<>")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("controlGroup")
            .withValue("cgroup")
            .withOperator("like")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("minPeakMemory")
            .withValue("13")
            .withOperator("=")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("maxPeakMemory")
            .withValue("14")
            .withOperator("<>")
    );
    listbodyConditions.add(
        new ListQueriesCondition()
            .withField("averagePeakMemory")
            .withValue("15")
            .withOperator(">")
    );

```

```
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("memorySkewPercent")
        .withValue("16")
        .withOperator("<")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("estimateMemory")
        .withValue("17")
        .withOperator(">=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("minDnTime")
        .withValue("18")
        .withOperator("<=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("maxDnTime")
        .withValue("19")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("averageDnTime")
        .withValue("20")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("dntimeSkewPercent")
        .withValue("21")
        .withOperator(">")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("warning")
        .withValue("告警")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("averagePeaklops")
        .withValue("22")
        .withOperator("<>")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("iopsSkewPercent")
        .withValue("23")
        .withOperator(">")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("wlmStatus")
        .withValue("语句运行状态")
        .withOperator("=")
);
listbodyConditions.add(
    new ListQueriesCondition()
        .withField("wlmAttrib")
        .withValue("语句属性")
        .withOperator("not like")
);
body.withTarget("db_queries");
body.withOrderBy("duration asc");
```

```
body.withConditions(listbodyConditions);
body.withLimit("1");
body.withOffset("0");
body.withProjectId("project_id");
body.withClusterId("cluster_id");
request.withBody(body);
try {
    ListQueriesResponse response = client.listQueries(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListQueriesRequest()
        listConditionsbody = [
            ListQueriesCondition(
                field="userName",
                value="用户名称",
                operator="="
            ),
            ListQueriesCondition(
                field="applicationName",
                value="应用名称",
                operator="<>"
            ),
            ListQueriesCondition(
                field="dbName",
                value="数据库名称",
                operator="="
            ),
            ListQueriesCondition(
                field="resourcePool",
                value="资源池",
```

```

        operator="<>"
    ),
    ListQueriesCondition(
        field="queryStatus",
        value="查询状态",
        operator="="
    ),
    ListQueriesCondition(
        field="enqueue",
        value="排队状态",
        operator="<>"
    ),
    ListQueriesCondition(
        field="lane",
        value="快慢车道",
        operator="="
    ),
    ListQueriesCondition(
        field="instName",
        value="接入CN",
        operator="<>"
    ),
    ListQueriesCondition(
        field="pid",
        value="会话ID",
        operator="="
    ),
    ListQueriesCondition(
        field="blockTime",
        value="1",
        operator="="
    ),
    ListQueriesCondition(
        field="duration",
        value="2",
        operator="<>"
    ),
    ListQueriesCondition(
        field="minCpuTime",
        value="3",
        operator=">"
    ),
    ListQueriesCondition(
        field="maxCpuTime",
        value="4",
        operator="<"
    ),
    ListQueriesCondition(
        field="totalCpuTime",
        value="5",
        operator=">="
    ),
    ListQueriesCondition(
        field="cpuSkewPercent",
        value="6",
        operator="<="
    ),
    ListQueriesCondition(
        field="spillInfo",
        value="dn下盘信息",
        operator="="
    ),
    ListQueriesCondition(
        field="minSpillSize",
        value="7",
        operator="<>"
    ),
    ListQueriesCondition(
        field="maxSpillSize",

```

```
        value="8",
        operator=">"
    ),
    ListQueriesCondition(
        field="averageSpillSize",
        value="9",
        operator="<"
    ),
    ListQueriesCondition(
        field="spillSkewPercent",
        value="10",
        operator=">="
    ),
    ListQueriesCondition(
        field="queryBand",
        value="作业类型",
        operator="<>"
    ),
    ListQueriesCondition(
        field="jobName",
        value="任务名称",
        operator="="
    ),
    ListQueriesCondition(
        field="jobInst",
        value="任务实例",
        operator="<>"
    ),
    ListQueriesCondition(
        field="clientHostname",
        value="主机名称",
        operator="="
    ),
    ListQueriesCondition(
        field="clientPort",
        value="TCP端口",
        operator="<>"
    ),
    ListQueriesCondition(
        field="waiting",
        value="是否等待",
        operator="="
    ),
    ListQueriesCondition(
        field="estimateTotalTime",
        value="11",
        operator="="
    ),
    ListQueriesCondition(
        field="estimateLeftTime",
        value="12",
        operator="<>"
    ),
    ListQueriesCondition(
        field="controlGroup",
        value="cgroup",
        operator="like"
    ),
    ListQueriesCondition(
        field="minPeakMemory",
        value="13",
        operator="="
    ),
    ListQueriesCondition(
        field="maxPeakMemory",
        value="14",
        operator="<>"
    ),
    ListQueriesCondition(
```

```
        field="averagePeakMemory",
        value="15",
        operator=">"
    ),
    ListQueriesCondition(
        field="memorySkewPercent",
        value="16",
        operator="<"
    ),
    ListQueriesCondition(
        field="estimateMemory",
        value="17",
        operator=">="
    ),
    ListQueriesCondition(
        field="minDnTime",
        value="18",
        operator="<="
    ),
    ListQueriesCondition(
        field="maxDnTime",
        value="19",
        operator="="
    ),
    ListQueriesCondition(
        field="averageDnTime",
        value="20",
        operator="<>"
    ),
    ListQueriesCondition(
        field="dntimeSkewPercent",
        value="21",
        operator=">"
    ),
    ListQueriesCondition(
        field="warning",
        value="告警",
        operator="="
    ),
    ListQueriesCondition(
        field="averagePeaklops",
        value="22",
        operator="<>"
    ),
    ListQueriesCondition(
        field="iopsSkewPercent",
        value="23",
        operator=">"
    ),
    ListQueriesCondition(
        field="wlmStatus",
        value="语句运行状态",
        operator="="
    ),
    ListQueriesCondition(
        field="wlmAttrib",
        value="语句属性",
        operator="not like"
    )
]
request.body = ListQueriesRequestBody(
    target="db_queries",
    order_by="duration asc",
    conditions=listConditionsbody,
    limit="1",
    offset="0",
    project_id="project_id",
    cluster_id="cluster_id"
)
```

```
response = client.list_queries(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListQueriesRequest{}
    var listConditionsbody = []model.ListQueriesCondition{
        {
            Field: "userName",
            Value: "用户名称",
            Operator: "=",
        },
        {
            Field: "applicationName",
            Value: "应用名称",
            Operator: "<>",
        },
        {
            Field: "dbName",
            Value: "数据库名称",
            Operator: "=",
        },
        {
            Field: "resourcePool",
            Value: "资源池",
            Operator: "<>",
        },
        {
            Field: "queryStatus",
            Value: "查询状态",
            Operator: "=",
        },
        {
            Field: "enqueue",
```



```

Value: "排队状态",
Operator: "<>",
},
{
Field: "lane",
Value: "快慢车道",
Operator: "=",
},
{
Field: "instName",
Value: "接入CN",
Operator: "<>",
},
{
Field: "pid",
Value: "会话ID",
Operator: "=",
},
{
Field: "blockTime",
Value: "1",
Operator: "=",
},
{
Field: "duration",
Value: "2",
Operator: "<>",
},
{
Field: "minCpuTime",
Value: "3",
Operator: ">",
},
{
Field: "maxCpuTime",
Value: "4",
Operator: "<",
},
{
Field: "totalCpuTime",
Value: "5",
Operator: ">=",
},
{
Field: "cpuSkewPercent",
Value: "6",
Operator: "<=",
},
{
Field: "spillInfo",
Value: "dn下盘信息",
Operator: "=",
},
{
Field: "minSpillSize",
Value: "7",
Operator: "<>",
},
{
Field: "maxSpillSize",
Value: "8",
Operator: ">",
},
{
Field: "averageSpillSize",
Value: "9",
Operator: "<",
},
{

```

```
Field: "spillSkewPercent",
Value: "10",
Operator: ">=",
},
{
Field: "queryBand",
Value: "作业类型",
Operator: "<>",
},
{
Field: "jobName",
Value: "任务名称",
Operator: "=",
},
{
Field: "jobInst",
Value: "任务实例",
Operator: "<>",
},
{
Field: "clientHostname",
Value: "主机名称",
Operator: "=",
},
{
Field: "clientPort",
Value: "TCP端口",
Operator: "<>",
},
{
Field: "waiting",
Value: "是否等待",
Operator: "=",
},
{
Field: "estimateTotalTime",
Value: "11",
Operator: "=",
},
{
Field: "estimateLeftTime",
Value: "12",
Operator: "<>",
},
{
Field: "controlGroup",
Value: "cgroup",
Operator: "like",
},
{
Field: "minPeakMemory",
Value: "13",
Operator: "=",
},
{
Field: "maxPeakMemory",
Value: "14",
Operator: "<>",
},
{
Field: "averagePeakMemory",
Value: "15",
Operator: ">",
},
{
Field: "memorySkewPercent",
Value: "16",
Operator: "<",
},
},
```

```

    {
      Field: "estimateMemory",
      Value: "17",
      Operator: ">=",
    },
    {
      Field: "minDnTime",
      Value: "18",
      Operator: "<=",
    },
    {
      Field: "maxDnTime",
      Value: "19",
      Operator: "=",
    },
    {
      Field: "averageDnTime",
      Value: "20",
      Operator: "<>",
    },
    {
      Field: "dntimeSkewPercent",
      Value: "21",
      Operator: ">",
    },
    {
      Field: "warning",
      Value: "告警",
      Operator: "=",
    },
    {
      Field: "averagePeaklops",
      Value: "22",
      Operator: "<>",
    },
    {
      Field: "iopsSkewPercent",
      Value: "23",
      Operator: ">",
    },
    {
      Field: "wlmStatus",
      Value: "语句运行状态",
      Operator: "=",
    },
    {
      Field: "wlmAttrib",
      Value: "语句属性",
      Operator: "not like",
    },
  },
}
orderByListQueriesRequestBody:= "duration asc"
request.Body = &model.ListQueriesRequestBody{
  Target: "db_queries",
  OrderBy: &orderByListQueriesRequestBody,
  Conditions: listConditionsbody,
  Limit: "1",
  Offset: "0",
  ProjectId: "project_id",
  ClusterId: "cluster_id",
}
response, err := client.ListQueries(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SQL列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.8 查询 SQL 执行信息

功能介绍

该接口用于查询SQL执行信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/dms/queries/{query_id}

表 5-512 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
query_id	是	String	查询ID。

表 5-513 Query 参数

参数	是否必选	参数类型	描述
ctime	否	Long	采集时间。

请求参数

无

响应参数

状态码： 200

表 5-514 响应 Body 参数

参数	参数类型	描述
code	Integer	响应码。
msg	String	响应信息。
data	ListQueriesDto object	响应数据。

表 5-515 ListQueriesDto

参数	参数类型	描述
virtual_cluster_id	Integer	虚拟集群ID
ctime	Long	采集时间
pid	String	会话id。
inst_name	String	实例名称。
waiting	Boolean	如果后台当前正等待锁则为true。
enqueue	String	工作负载管理资源状态。
warning	String	主要显示如下几类告警信息以及sql自诊断调优相关告警。
query	String	查询语句。
lane	String	快慢车道 (fast or slow)。
db_name	String	数据库名称。
priority	String	job在资源池中的优先级，取值： 1,2,4,8 (rush、high、medium、low)。

参数	参数类型	描述
query_id	String	语句执行使用的内部query_id。
query_band	String	用于标示作业类型，可通过guc参数query_band进行设置，默认为空字符串。
job_name	String	这个值是从query_band的字段中取出来的，位置0。
job_inst	String	这个值是从query_band的字段中取出来的，位置1。
user_name	String	连接到后端的用户名。
application_name	String	连接到后端的应用名。
client_address	String	连接到后端的客户端的ip地址。
client_hostname	String	客户端的主机名。
client_port	String	客户端用于与后端通讯的tcp端口号。
start_time	Long	语句执行的开始时间。
block_time	Long	语句执行前的阻塞时间（单位ms）。
duration	Long	语句已经执行的时间（单位ms）。
estimate_total_time	Long	语句执行预估总时间（单位ms）。
estimate_left_time	Long	语句执行预估剩余时间（单位ms）。
resource_pool	String	用户使用的资源池。
control_group	String	语句所使用的cgroup。
min_peak_memory	Integer	语句在所有dn上的最小内存峰值（单位mb）。
max_peak_memory	Integer	语句在所有dn上的最大内存峰值（单位mb）。
average_peak_memory	Integer	语句执行过程中的内存使用平均值（单位mb）。
memory_skew_percent	Integer	语句在各dn间的内存使用倾斜率。
estimate_memory	Integer	语句预估使用内存（单位mb）。
spill_info	String	语句在所有dn上的下盘信息。

参数	参数类型	描述
min_spill_size	Integer	若发生下盘，所有dn上下盘的最小数据量 (单位mb) 默认为0。
max_spill_size	Integer	若发生下盘，所有dn上下盘的最大数据量 (单位mb) 默认为0。
average_spill_size	Integer	若发生下盘，所有dn上下盘的平均数据量 (单位mb) 默认为0。
spill_skew_percent	Integer	若发生下盘，dn间下盘倾斜率。
min_dn_time	Long	语句在所有dn上的最小执行时间 (单位ms)。
max_dn_time	Long	语句在所有dn上的最大执行时间 (单位ms)。
average_dn_time	Long	语句在所有dn上的平均执行时间 (单位ms)。
dn_time_skew_percent	Integer	语句在各dn间的执行时间倾斜率。
min_cpu_time	Long	语句在所有dn上的最小cpu时间 (单位ms)。
max_cpu_time	Long	语句在所有dn上的最大cpu时间 (单位ms)。
total_cpu_time	Long	语句在所有dn上的cpu总时间 (单位ms)。
cpu_skew_percent	Integer	语句在各dn间的cpu时间倾斜率。
average_peak_iops	Integer	语句在所有dn上的每秒平均io峰值 (列存单位是次/s, 行存单位是万次/s)。
iops_skew_percent	Integer	语句在dn间的io倾斜率。
max_peak_iops	Integer	语句在所有dn上的每秒最大io峰值 (列存单位是次/s, 行存单位是万次/s)。
min_peak_iops	Integer	语句在所有dn上的每秒最小io峰值 (列存单位是次/s, 行存单位是万次/s)。
query_plan	String	查询计划。
query_status	String	当前查询语句的实时运行状态 (active, idle, idle in transaction, idle in transaction(aborted), fastpath function call, disabled)。
wlm_status	String	当前查询语句在资源池上的运行状态 (pending, running, finished, aborted, active, unknown)。
wlm_attr	String	语句的属性 (ordinary, simple, complicated, internal)
system_query	Boolean	是否系统查询。

参数	参数类型	描述
backend_start	Long	该过程开始的时间，即当客户端连接服务器时。
elapsed_time	Long	到目前为止的执行时间。
curr_xact_start	Long	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
state_change	Long	上次状态改变的时间。
query_start	Long	语句执行的开始时间。
query_elapsed_time	Long	语句当前为止的实际执行时间(单位: s)。

请求示例

https://{Endpoint}/v1/{project_id}/clusters/{cluster_id}/dms/queries/{query_id}?ctime=1699062846000

响应示例

状态码： 200

```
{
  "code": 0,
  "msg": "OK",
  "data": {
    "ctime": 1699062846000,
    "pid": "140535026615872",
    "waiting": false,
    "duration": 0,
    "enqueue": "",
    "warning": "",
    "query": "WLM fetch collect info from data nodes",
    "lane": "",
    "priority": null,
    "virtual_cluster_id": 0,
    "inst_name": "cn_5002",
    "db_name": "postgres",
    "query_id": "14552256295985061",
    "query_band": "",
    "job_name": "",
    "job_inst": "",
    "user_name": "Ruby",
    "application_name": "workload",
    "client_address": "",
    "client_hostname": "",
    "client_port": "",
    "start_time": 0,
    "block_time": 0,
    "estimate_total_time": 0,
    "estimate_left_time": 0,
    "resource_pool": "default_pool",
    "control_group": "",
    "min_peak_memory": 0,
    "max_peak_memory": 0,
    "average_peak_memory": 0,
    "memory_skew_percent": 0,
    "estimate_memory": 0,
    "spill_info": "",
    "min_spill_size": 0,
  }
}
```



```
"max_spill_size" : 0,
"average_spill_size" : 0,
"spill_skew_percent" : 0,
"min_dn_time" : 0,
"max_dn_time" : 0,
"average_dn_time" : 0,
"dntime_skew_percent" : 0,
"min_cpu_time" : 0,
"max_cpu_time" : 0,
"total_cpu_time" : 0,
"cpu_skew_percent" : 0,
"average_peak_iops" : 0,
"iops_skew_percent" : 0,
"max_peak_iops" : 0,
"min_peak_iops" : 0,
"query_plan" : "",
"query_status" : "active",
"wlm_status" : "",
"wlm_attrib" : "",
"system_query" : false,
"backend_start" : 0,
"elapsed_time" : 0,
"curr_xact_start" : 0,
"state_change" : 0,
"query_start" : 0,
"query_elapsed_time" : 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowQueryDetailSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowQueryDetailRequest request = new ShowQueryDetailRequest();
        request.withCtime(<ctime>L);
    }
}
```

```

try {
    ShowQueryDetailResponse response = client.showQueryDetail(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowQueryDetailRequest()
        request.ctime = <ctime>
        response = client.show_query_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowQueryDetailRequest{}
ctimeRequest:= int64(<ctime>)
request.Ctime = &ctimeRequest
response, err := client.ShowQueryDetail(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SQL执行信息成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.9 查询集群使用指标列表

功能介绍

该接口用于查询集群使用指标列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/dms/metrics

表 5-516 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-517 Query 参数

参数	是否必选	参数类型	描述
offset	是	Integer	偏移量，表示从此偏移量开始查询，offset>=0。
limit	是	Integer	每页显示的条目数量，最大1000。
order_by	否	String	排序字段。固定取值。 <ul style="list-style-type: none"> create_time: 创建时间。
sort_by	否	String	正序还是倒叙。固定取值。 <ul style="list-style-type: none"> asc: 正序。 desc: 倒序。

请求参数

无

响应参数

状态码： 200

表 5-518 响应 Body 参数

参数	参数类型	描述
code	Integer	响应码。
msg	String	响应信息。
data	Array of ClusterMetric objects	指标列表。

参数	参数类型	描述
count	Long	总列表大小。

表 5-519 ClusterMetric

参数	参数类型	描述
scope	String	指标名称。
fields	Array of SimpleFieldDto objects	指标表相关字段信息。
metric_name	String	作用域。
collect_rate	Integer	采集速率。
collect_range	Array of strings	采集时间范围。
create_time	String	创建时间。

表 5-520 SimpleFieldDto

参数	参数类型	描述
column_name	String	指标表对应字段名称。
column_type	String	指标表对应字段类型。

请求示例

```
https://{Endpoint}/v1/{project_id}/clusters/{cluster_id}/dms/metrics?
limit=1&offset=0&order_by=create_time&sort_by=desc
```

响应示例

状态码： 200

```
{
  "code": 0,
  "msg": "OK",
  "data": [ {
    "scope": "CN,ANY",
    "fields": [ {
      "column_name": "db_name",
      "column_type": "text"
    }, {
      "column_name": "schema_name",
      "column_type": "text"
    }, {
      "column_name": "table_name",
      "column_type": "text"
    }, {

```

```

"column_name": "table_owner",
"column_type": "text"
}, {
"column_name": "table_size",
"column_type": "bigint"
}, {
"column_name": "dirty_page_rate",
"column_type": "double precision"
}],
"metric_name": "TableDirtyPageRate",
"collect_rate": 7200,
"collect_range": [],
"create_time": "1701915436000"
}],
"count": 1
}
    
```

状态码

状态码	描述
200	查询集群使用指标列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.14.10 获取指定指标相关采集数据

功能介绍

该接口用于获取指定指标相关采集数据。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1/{project_id}/clusters/{cluster_id}/dms/metrics/{metric_name}
```

表 5-521 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
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
metric_name	是	String	指标名称。

表 5-522 Query 参数

参数	是否必选	参数类型	描述
offset	是	Integer	偏移量，表示从此偏移量开始查询，offset>=0。
limit	是	Integer	每页显示的条目数量，最大1000。
from	是	Long	采集开始时间，13位时间戳。
to	是	Long	采集结束时间，13位时间戳。开始时间到结束时间最多不超过一天。
order_by	否	String	排序字段。固定取值。 <ul style="list-style-type: none"> ctime: 采集时间。
sort_by	否	String	正序还是倒叙。固定取值。 <ul style="list-style-type: none"> asc: 正序。 desc: 倒序。

请求参数

无

响应参数

状态码： 200

表 5-523 响应 Body 参数

参数	参数类型	描述
code	Integer	响应码。
msg	String	响应信息。
data	Array of Map<String,String> objects	指标采集数据列表。详情请参见 自定义指标列表 。
count	Long	总列表大小。

请求示例

```
https://{Endpoint}/v1/{project_id}/clusters/{cluster_id}/dms/metrics/TableDirtyPageRate?limit=1&offset=0&from=1701914324000&to=1701942194000&order_by=ctime&sort_by=desc
```

响应示例

状态码： 200

查询指标采集数据对象

```
{
  "code" : 0,
  "msg" : "OK",
  "data" : [ {
    "table_size" : 40960,
    "virtual_cluster_id" : 53,
    "db_name" : "postgres",
    "ctime" : 1701929836000,
    "table_owner" : "Ruby",
    "schema_name" : "pmk",
    "table_name" : "pmk_meta_data",
    "dirty_page_rate" : 0.8
  } ],
  "count" : 3
}
```

状态码

状态码	描述
200	查询指定指标采集数据对象成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15 数据库权限管理

5.15.1 同步 IAM 用户到数据库

功能介绍

该接口用于同步IAM用户到数据库。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/clusters/{cluster_id}/db-manager/sync-iam-user
```

表 5-524 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-525 响应 Body 参数

参数	参数类型	描述
sync_user	Array of strings	创建成功用户列表。

请求示例

```
POST https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/sync-iam-user
```

响应示例

状态码： 200

同步iam用户响应信息，成功创建用户列表：

```
{  
  "sync_user": [ "user1", "user2" ]  
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class SynclamUsersSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        SynclamUsersRequest request = new SynclamUsersRequest();
        try {
            SynclamUsersResponse response = client.synclamUsers(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

```

```
credentials = BasicCredentials(ak, sk) \

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

try:
    request = SynclamUsersRequest()
    response = client.sync_iam_users(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SynclamUsersRequest{}
    response, err := client.SynclamUsers(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	同步IAM用户响应信息，成功创建用户列表。
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.2 查询所有数据库用户/角色

功能介绍

该接口用于查询所有数据库用户/角色。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/db-manager/users

表 5-526 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

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
users	Array of DatabaseUser objects	用户列表。

表 5-528 DatabaseUser

参数	参数类型	描述
name	String	用户名。
login	Boolean	是否可以登录。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/users
```

响应示例

状态码： 200

用户/角色列表

```
{
  "users": [ {
    "name": "user1",
    "login": true
  }, {
    "name": "user2",
    "login": false
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListDatabaseUsersSolution {
    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
ListDatabaseUsersRequest request = new ListDatabaseUsersRequest();
try {
    ListDatabaseUsersResponse response = client.listDatabaseUsers(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDatabaseUsersRequest()
        response = client.list_database_users(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDatabaseUsersRequest{}
    response, err := client.ListDatabaseUsers(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	用户/角色列表
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.3 查询指定用户信息

功能介绍

该接口用于查询指定用户信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/db-manager/users/{name}

表 5-529 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
name	是	String	用户名。

请求参数

无

响应参数

状态码： 200

表 5-530 响应 Body 参数

参数	参数类型	描述
name	String	用户名称。
login	Boolean	是否可以登录。
creatorole	Boolean	创建角色权限。
createdb	Boolean	创建数据库权限。
systemadmin	Boolean	系统管理员。
auditadmin	Boolean	审计管理员。
inherit	Boolean	继承所在组权限。
useft	Boolean	访问外表权限。

参数	参数类型	描述
conn_limit	Integer	连接数限制。
replication	Boolean	是否允许流复制。
valid_begin	Long	角色生效时间。
valid_until	Long	角色过期时间。
lock	Boolean	是否锁定。
desc	String	描述。
user_type	String	用户类型。
logical_cluster	String	所属逻辑集群。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/users/user1
```

响应示例

状态码： 200

用户详细信息

```
{
  "name": "user1",
  "login": true,
  "creatorole": true,
  "createdb": true,
  "systemadmin": false,
  "auditadmin": false,
  "inherit": false,
  "useft": true,
  "conn_limit": -1,
  "replication": false,
  "valid_begin": 1681971166000,
  "valid_until": 1681971166000,
  "lock": false,
  "desc": "",
  "user_type": "",
  "logical_cluster": ""
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowDatabaseUserSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDatabaseUserRequest request = new ShowDatabaseUserRequest();
        try {
            ShowDatabaseUserResponse response = client.showDatabaseUser(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowDatabaseUserRequest()
```

```
response = client.show_database_user(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDatabaseUserRequest{}
    response, err := client.ShowDatabaseUser(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	用户详细信息
400	请求错误
401	鉴权失败。
403	没有操作权限。

状态码	描述
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.4 修改指定用户信息

功能介绍

该接口用于修改指定用户信息。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/db-manager/users/{name}

表 5-531 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。
name	是	String	用户名。

请求参数

表 5-532 请求 Body 参数

参数	是否必选	参数类型	描述
login	否	Boolean	是否可以登录。
creatorole	否	Boolean	创建角色权限。
createdb	否	Boolean	创建数据库权限。
systemadmin	否	Boolean	系统管理员。
auditadmin	否	Boolean	审计管理员
inherit	否	Boolean	继承所在组权限。

参数	是否必选	参数类型	描述
useft	否	Boolean	访问外表权限。
conn_limit	否	Integer	连接数限制。
replication	否	Boolean	是否允许流复制。
valid_begin	否	String	角色生效时间 yyyy-MM-ddTHH:mm:ssZ。
valid_until	否	String	角色过期时间 yyyy-MM-ddTHH:mm:ssZ。
lock	否	Boolean	是否锁定。

响应参数

无

请求示例

修改集群ID为“a89aea88-7ea2-40bd-8ac8-8b93e169e5d6”下的用户user1的用户信息，即可以登录数据库并拥有创建角色、创建数据库、访问外表权限，但不赋予给用户系统管理员和审计管理员权限，并不限制连接数等：

```
POST https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/users/user1
```

```
{
  "login": true,
  "creatorole": true,
  "createdb": true,
  "systemadmin": false,
  "auditadmin": false,
  "inherit": false,
  "useft": true,
  "conn_limit": -1,
  "replication": false,
  "valid_begin": "2023-04-03T12:23:01",
  "valid_until": "2023-04-18T12:23:01",
  "lock": false
}
```

响应示例

状态码： 200

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class UpdateDatabaseUserInfoSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        UpdateDatabaseUserInfoRequest request = new UpdateDatabaseUserInfoRequest();
        DatabaseUserInfoReq body = new DatabaseUserInfoReq();
        body.withLock(false);
        body.withValidUntil("2023-04-18T12:23:01");
        body.withValidBegin("2023-04-03T12:23:01");
        body.withReplication(false);
        body.withConnLimit(-1);
        body.withUseft(true);
        body.withInherit(false);
        body.withAuditadmin(false);
        body.withSystemadmin(false);
        body.withCreatedb(true);
        body.withCreatorole(true);
        body.withLogin(true);
        request.withBody(body);
        try {
            UpdateDatabaseUserInfoResponse response = client.updateDatabaseUserInfo(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = UpdateDatabaseUserInfoRequest()
        request.body = DatabaseUserInfoReq(
            lock=False,
            valid_until="2023-04-18T12:23:01",
            valid_begin="2023-04-03T12:23:01",
            replication=False,
            conn_limit=-1,
            useft=True,
            inherit=False,
            auditadmin=False,
            systemadmin=False,
            createdb=True,
            createrole=True,
            login=True
        )
        response = client.update_database_user_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
```

```

        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build()

    request := &model.UpdateDatabaseUserInfoRequest{
        lockDatabaseUserInfoReq:= false
        validUntilDatabaseUserInfoReq:= "2023-04-18T12:23:01"
        validBeginDatabaseUserInfoReq:= "2023-04-03T12:23:01"
        replicationDatabaseUserInfoReq:= false
        connLimitDatabaseUserInfoReq:= int32(-1)
        useftDatabaseUserInfoReq:= true
        inheritDatabaseUserInfoReq:= false
        auditadminDatabaseUserInfoReq:= false
        systemadminDatabaseUserInfoReq:= false
        createdbDatabaseUserInfoReq:= true
        createroleDatabaseUserInfoReq:= true
        loginDatabaseUserInfoReq:= true
        request.Body = &model.DatabaseUserInfoReq{
            Lock: &lockDatabaseUserInfoReq,
            ValidUntil: &validUntilDatabaseUserInfoReq,
            ValidBegin: &validBeginDatabaseUserInfoReq,
            Replication: &replicationDatabaseUserInfoReq,
            ConnLimit: &connLimitDatabaseUserInfoReq,
            Useft: &useftDatabaseUserInfoReq,
            Inherit: &inheritDatabaseUserInfoReq,
            Auditadmin: &auditadminDatabaseUserInfoReq,
            Systemadmin: &systemadminDatabaseUserInfoReq,
            Createdb: &createdbDatabaseUserInfoReq,
            Createrole: &createroleDatabaseUserInfoReq,
            Login: &loginDatabaseUserInfoReq,
        }
    }
    response, err := client.UpdateDatabaseUserInfo(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	用户详细信息
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.5 修改数据库对象权限

功能介绍

该接口用于修改数据库对象权限。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/db-manager/authority

表 5-533 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-534 请求 Body 参数

参数	是否必选	参数类型	描述
type	是	String	对象类型 [DATABASE SCHEMA TABLE VIEW COLUMN FUNCTION SEQUENCE NODEGROUP ROLE]。
is_grant	是	Boolean	是否授权操作。
grant_list	否	Array of Grant objects	授权列表 is_grant为true时必填。
revoke_list	否	Array of Revoke objects	撤销权限列表 is_grant为false时必填。
role_list	是	Array of strings	被授权角色列表。
object_list	是	Array of strings	权限所属对象列表。
all_object	否	Boolean	schema下所有数据库对象权限，默认false。

参数	是否必选	参数类型	描述
cascade	否	Boolean	撤销权限是否级联撤销，默认为 true。 缺省值： true
database	是	String	数据库名称。
schema	否	String	模式名称。
table	否	String	表名。

表 5-535 Grant

参数	是否必选	参数类型	描述
permission	是	String	<p>权限名称，根据不同数据库对象类型，拥有权限不同：</p> <ul style="list-style-type: none"> • database CREATE CONNECT TEMPORARY TEMP ALL PRIVILEGES • schema CREATE USAGE ALTER DROP ALL PRIVILEGES • table SELECT INSERT UPDATE DELETE TRUNCATE REFERENCES TRIGGER ANALYZE ANALYSE VACUUM ALTER DROP ALL PRIVILEGES • view SELECT INSERT UPDATE DELETE TRUNCATE REFERENCES TRIGGER ANALYZE ANALYSE VACUUM ALTER DROP ALL PRIVILEGES • column SELECT INSERT UPDATE REFERENCES ALL PRIVILEGES • function EXECUTE ALL PRIVILEGES • sequence SELECT UPDATE USAGE ALL PRIVILEGES • nodegroup CREATE USAGE COMPUTE ALL PRIVILEGES • role role_name (角色名称)
grant_with	是	Boolean	是否包含授权选项。

表 5-536 Revoke

参数	是否必选	参数类型	描述
permission	是	String	<p>权限名称，根据不通数据库对象类型，拥有权限不同：</p> <ul style="list-style-type: none"> • database CREATE CONNECT TEMPORARY TEMP ALL PRIVILEGES • schema CREATE USAGE ALTER DROP ALL PRIVILEGES • table SELECT INSERT UPDATE DELETE TRUNCATE REFERENCES TRIGGER ANALYZE ANALYSE VACUUM ALTER DROP ALL PRIVILEGES • view SELECT INSERT UPDATE DELETE TRUNCATE REFERENCES TRIGGER ANALYZE ANALYSE VACUUM ALTER DROP ALL PRIVILEGES • column SELECT INSERT UPDATE REFERENCES ALL PRIVILEGES • function EXECUTE ALL PRIVILEGES • sequence SELECT UPDATE USAGE ALL PRIVILEGES • nodegroup CREATE USAGE COMPUTE ALL PRIVILEGES • role role_name (角色名称)
revoke_with	是	Boolean	是否仅移除授权选项。

响应参数

状态码： 200

表 5-537 响应 Body 参数

参数	参数类型	描述
view_sql	Array of strings	sql列表。

请求示例

修改集群ID为“a89aea88-7ea2-40bd-8ac8-8b93e169e5d6”的数据库对象权限，即对gaussdb数据库下表table1和table2进行“SELECT”授权，授权角色为用户1和用户2，模式为public：

```
POST https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/authority
```

```
{
  "type": "table",
  "is_grant": true,
  "grant_list": [ {
    "permission": "SELECT",
    "grant_with": true
  } ],
  "role_list": [ "user1", "user2" ],
  "object_list": [ "table1", "table2" ],
  "database": "gaussdb",
  "schema": "public"
}
```

响应示例

状态码：200

用户详细信息

```
{
  "view_sql": [ "GRANT SELECT ON TABLE public.test_t1 TO user1 WITH GRANT OPTION;", "GRANT UPDATE ON TABLE public.test_t1 TO user1 ;" ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class UpdateDatabaseAuthoritySolution {
```

```
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);

    DwsClient client = DwsClient.newBuilder()
        .withCredential(auth)
        .withRegion(DwsRegion.valueOf("cn-north-4"))
        .build();
    UpdateDatabaseAuthorityRequest request = new UpdateDatabaseAuthorityRequest();
    DatabasePermissionReq body = new DatabasePermissionReq();
    List<String> listbodyRoleList = new ArrayList<>();
    listbodyRoleList.add("user1");
    listbodyRoleList.add("user2");
    List<Grant> listbodyGrantList = new ArrayList<>();
    listbodyGrantList.add(
        new Grant()
            .withPermission("SELECT")
            .withGrantWith(true)
    );
    body.withSchema("public");
    body.withDatabase("gaussdb");
    body.withObjectList("[table1, table2]");
    body.withRoleList(listbodyRoleList);
    body.withGrantList(listbodyGrantList);
    body.withIsGrant(true);
    body.withType("table");
    request.withBody(body);
    try {
        UpdateDatabaseAuthorityResponse response = client.updateDatabaseAuthority(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

client = DwsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(DwsRegion.value_of("cn-north-4")) \
    .build()

try:
    request = UpdateDatabaseAuthorityRequest()
    listRoleListbody = [
        "user1",
        "user2"
    ]
    listGrantListbody = [
        Grant(
            permission="SELECT",
            grant_with=True
        )
    ]
    request.body = DatabasePermissionReq(
        schema="public",
        database="gaussdb",
        object_list="[table1, table2]",
        role_list=listRoleListbody,
        grant_list=listGrantListbody,
        is_grant=True,
        type="table"
    )
    response = client.update_database_authority(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
```

```

Build()

request := &model.UpdateDatabaseAuthorityRequest{}
var listRoleListbody = []string{
    "user1",
    "user2",
}
var listGrantListbody = []model.Grant{
    {
        Permission: "SELECT",
        GrantWith: true,
    },
}
schemaDatabasePermissionReq:= "public"
request.Body = &model.DatabasePermissionReq{
    Schema: &schemaDatabasePermissionReq,
    Database: "gaussdb",
    ObjectList: "[table1, table2]",
    RoleList: listRoleListbody,
    GrantList: &listGrantListbody,
    IsGrant: true,
    Type: "table",
}
response, err := client.UpdateDatabaseAuthority(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	用户详细信息
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.6 查询数据库对象权限

功能介绍

该接口用于查询数据库对象权限。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/db-manager/authority

表 5-538 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-539 Query 参数

参数	是否必选	参数类型	描述
type	是	String	对象类型 [DATABASE SCHEMA TABLE VIEW COLUMN FUNCTION SEQUENCE NODEGROUP]
name	是	Array of strings	对象名称。
database	是	String	数据库名。
schema	否	String	模式名，对象类型为TABLE、VIEW、COLUMN、FUNCTION、SEQUENCE时必选。
table	否	String	表名，对象类型为COLUMN时必选。

请求参数

无

响应参数

状态码： 200

表 5-540 响应 Body 参数

参数	参数类型	描述
authorities	Array of ObjectAuthority objects	对象权限集合。

表 5-541 ObjectAuthority

参数	参数类型	描述
name	String	对象名称。
role_authority	Array of RoleAuthority objects	角色权限集合。

表 5-542 RoleAuthority

参数	参数类型	描述
role	String	角色名称。
right_list	Array of strings	权限列表。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/
a89aea88-7ea2-40bd-8ac8-8b93e169e5d6/db-manager/authority?
type=table&name=table1&database=gaussdb&schema=public
```

响应示例

状态码： 200

数据库对象权限信息

```
{
  "authorities" : [ {
    "name" : "table1",
    "role_authority" : [ {
      "role" : "role1",
      "right_list" : [ "SELECT", "UPDATE" ]
    }, {
      "role" : "role2",
      "right_list" : [ "SELECT", "UPDATE" ]
    } ]
  } ]
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

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

public class ShowDatabaseAuthoritySolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDatabaseAuthorityRequest request = new ShowDatabaseAuthorityRequest();
        request.withType("<type>");
        request.withName();
        request.withDatabase("<database>");
        request.withSchema("<schema>");
        request.withTable("<table>");
        try {
            ShowDatabaseAuthorityResponse response = client.showDatabaseAuthority(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowDatabaseAuthorityRequest()
        request.type = "<type>"
        request.name =
        request.database = "<database>"
        request.schema = "<schema>"
        request.table = "<table>"
        response = client.show_database_authority(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDatabaseAuthorityRequest{}
    request.Type = "<type>"
    request.Database = "<database>"
```

```

schemaRequest:= "<schema>"
request.Schema = &schemaRequest
tableRequest:= "<table>"
request.Table = &tableRequest
response, err := client.ShowDatabaseAuthority(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	数据库对象权限信息
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.7 获得集群运维账户状态

功能介绍

该接口用于获得数据库运维账户状态。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/db-manager/om-user/status

表 5-543 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
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

状态码： 200

表 5-544 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。
om_user_info	DatabaseOmUserInfo object	运维用户信息。

表 5-545 DatabaseOmUserInfo

参数	参数类型	描述
om_user_status	String	运维账户状态。
om_user_expires_time	String	运维账户过期状态。

请求示例

获得运维用户状态

<https://cn-north-208-console.ulanhqab.huawei.com/dws/rest-dms/v1.0/05f2cff45100d5112f4bc00b794ea08e/clusters/e7107fab-cac0-4261-916b-7f32b3053034/db-manager/om-user/status>

```
{
  "error_code" : 0,
  "error_msg" : null,
  "om_user_info" : {
    "om_user_status" : "on",
    "om_user_expires_time" : 1684918485000
  }
}
```

响应示例

状态码： 200

集群运维用户状态信息响应成功。

```
{
  "error_code" : 0,
  "error_msg" : null,
  "om_user_info" : null
}
```

SDK 代码示例

SDK代码示例如下。

Java

获得运维用户状态

```
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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ShowDatabaseOmUserStatusSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ShowDatabaseOmUserStatusRequest request = new ShowDatabaseOmUserStatusRequest();
        try {
            ShowDatabaseOmUserStatusResponse response = client.showDatabaseOmUserStatus(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

获得运维用户状态

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ShowDatabaseOmUserStatusRequest()
        response = client.show_database_om_user_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

获得运维用户状态

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
```



```

    WithCredential(auth).
    Build()

    request := &model.ShowDatabaseOmUserStatusRequest{}
    response, err := client.ShowDatabaseOmUserStatus(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	集群运维用户响应
400	请求错误
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.15.8 执行运维用户操作

功能介绍

该接口用于进行数据库运维账户操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/db-manager/om-user/action

表 5-546 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
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-547 请求 Body 参数

参数	是否必选	参数类型	描述
operation	是	String	操作类型。取值如下： <ul style="list-style-type: none"> addOmUser：添加运维用户。 deleteOmUser：删除运维用户。 increaseOmUserPeriod：延长用户有效期。

响应参数

状态码： 200

表 5-548 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码。
error_msg	String	错误信息。
om_user_info	DatabaseOmUserInfo object	运维用户信息。

表 5-549 DatabaseOmUserInfo

参数	参数类型	描述
om_user_status	String	运维账户状态。
om_user_expires_time	String	运维账户过期状态。

请求示例

- 添加运维用户

```
https://cn-north-208-console.ulanhqab.huawei.com/dws/rest-dms/v1.0/05f2cff45100d5112f4bc00b794ea08e/clusters/e7107fab-cac0-4261-916b-7f32b3053034/db-manager/om-user/action
```

```
{  
  "operation" : "addOmUser"  
}
```

- 删除运维用户

```
https://cn-north-208-console.ulanhqab.huawei.com/dws/rest-dms/v1.0/05f2cff45100d5112f4bc00b794ea08e/clusters/e7107fab-cac0-4261-916b-7f32b3053034/db-manager/om-user/action
```

```
{  
  "operation" : "deleteOmUser"  
}
```

- 延长用户有效期

```
https://cn-north-208-console.ulanhqab.huawei.com/dws/rest-dms/v1.0/05f2cff45100d5112f4bc00b794ea08e/clusters/e7107fab-cac0-4261-916b-7f32b3053034/db-manager/om-user/action
```

```
{  
  "operation" : "increaseOmUserPeriod"  
}
```

响应示例

状态码： 200

执行集群运维用户操作成功。

```
{  
  "error_code" : 0,  
  "error_msg" : null,  
  "om_user_info" : null  
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 添加运维用户

```
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.dws.v2.region.DwsRegion;  
import com.huaweicloud.sdk.dws.v2.*;  
import com.huaweicloud.sdk.dws.v2.model.*;  
  
public class ExecuteDatabaseOmUserActionSolution {  
  
    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);

DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("cn-north-4"))
    .build();
ExecuteDatabaseOmUserActionRequest request = new ExecuteDatabaseOmUserActionRequest();
DatabaseOmUserActionReq body = new DatabaseOmUserActionReq();
body.withOperation("addOmUser");
request.withBody(body);
try {
    ExecuteDatabaseOmUserActionResponse response =
client.executeDatabaseOmUserAction(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());
}
}
```

- **删除运维用户**

```
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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ExecuteDatabaseOmUserActionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
```

```
ExecuteDatabaseOmUserActionRequest request = new ExecuteDatabaseOmUserActionRequest();
DatabaseOmUserActionReq body = new DatabaseOmUserActionReq();
body.withOperation("deleteOmUser");
request.withBody(body);
try {
    ExecuteDatabaseOmUserActionResponse response =
client.executeDatabaseOmUserAction(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());
}
}
```

- **延长用户有效期**

```
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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ExecuteDatabaseOmUserActionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ExecuteDatabaseOmUserActionRequest request = new ExecuteDatabaseOmUserActionRequest();
        DatabaseOmUserActionReq body = new DatabaseOmUserActionReq();
        body.withOperation("increaseOmUserPeriod");
        request.withBody(body);
        try {
            ExecuteDatabaseOmUserActionResponse response =
client.executeDatabaseOmUserAction(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

- 添加运维用户

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ExecuteDatabaseOmUserActionRequest()
        request.body = DatabaseOmUserActionReq(
            operation="addOmUser"
        )
        response = client.execute_database_om_user_action(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 删除运维用户

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(DwsRegion.value_of("cn-north-4")) \  
.build()  
  
try:  
    request = ExecuteDatabaseOmUserActionRequest()  
    request.body = DatabaseOmUserActionReq(  
        operation="deleteOmUser"  
    )  
    response = client.execute_database_om_user_action(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

- 延长用户有效期

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkdws.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")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = DwsClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(DwsRegion.value_of("cn-north-4")) \  
        .build()  
  
    try:  
        request = ExecuteDatabaseOmUserActionRequest()  
        request.body = DatabaseOmUserActionReq(  
            operation="increaseOmUserPeriod"  
        )  
        response = client.execute_database_om_user_action(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

- 添加运维用户

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteDatabaseOmUserActionRequest{}
    request.Body = &model.DatabaseOmUserActionReq{
        Operation: "addOmUser",
    }
    response, err := client.ExecuteDatabaseOmUserAction(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 删除运维用户

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteDatabaseOmUserActionRequest{}
    request.Body = &model.DatabaseOmUserActionReq{
        Operation: "deleteOmUser",
    }
    response, err := client.ExecuteDatabaseOmUserAction(request)
```



```

if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
}

```

- 延长用户有效期

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteDatabaseOmUserActionRequest{}
    request.Body = &model.DatabaseOmUserActionReq{
        Operation: "increaseOmUserPeriod",
    }
    response, err := client.ExecuteDatabaseOmUserAction(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	集群运维用户响应
400	请求错误
401	鉴权失败。

状态码	描述
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.16 日志管理

5.16.1 获取 LTS 日志列表

功能介绍

该接口用于获取LTS日志列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/lts-logs

表 5-550 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-551 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数。
offset	否	Integer	偏移量。

请求参数

无

响应参数

状态码： 200

表 5-552 响应 Body 参数

参数	参数类型	描述
access_status	String	日志开启状态。
lts_access_list	Array of LtslogInfo objects	LTS日志列表。
count	Integer	总数量。

表 5-553 LtslogInfo

参数	参数类型	描述
status	String	配置状态，有效值包括： <ul style="list-style-type: none"> • OPEN：开启中。 • CLOSE：已关闭。
id	String	日志ID。
log_type	String	日志类型。
log_desc	String	日志描述。
access_url	String	LTS日志访问URL。

请求示例

```
GET https://{Endpoint}/v2/89cd04f168b84af6be287f71730fdb4b/clusters/e59d6b86-9072-46eb-a996-13f8b44994c1/lts-logs
```

响应示例

状态码： 200

获取LTS日志列表成功。

```
{
  "access_status": "OPEN",
  "lts_access_list": [ {
    "status": "OPEN",
    "id": "c0c4e5f2-9b2a-4b47-a649-baf40b33e2e0",
    "log_type": "messages",
    "log_desc": "operating system messages log",
    "access_url": "/lts/?region=cn-north-7&locale=#/cts/logEventsLeftMenu/events?groupId=b6680a92-e14f-4a7d-b669-4f702db806f7&groupName=z30024644-s-5&topicId=1a9fe6d0-d383-4d58-adb6-2c26d229944e&topicName=messages&epsId=0"
  } ],
  "count": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListLtsLogsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        ListLtsLogsRequest request = new ListLtsLogsRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        try {
            ListLtsLogsResponse response = client.listLtsLogs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListLtsLogsRequest()
    request.limit = <limit>
    request.offset = <offset>
    response = client.list_lts_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListLtsLogsRequest{}
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    response, err := client.ListLtsLogs(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LTS日志列表成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
404	找不到资源。
500	服务内部错误。
503	服务不可用。

5.16.2 开启云服务日志

功能介绍

该接口用于开启集群LTS云日志服务。

调用方法

请参见[如何调用API](#)。

URI

```
POST /v1/{project_id}/clusters/{cluster_id}/lts-logs/enable
```

表 5-554 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

无

请求示例

```
https://{Endpoint}/v2/05f2cff45100d5112f4bc00b794ea08e/clusters/a862e17c-2981-4150-9018-f9a40511999c/lts-logs/enable
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class EnableLtsLogsSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
            .build();
        EnableLtsLogsRequest request = new EnableLtsLogsRequest();
        try {
            EnableLtsLogsResponse response = client.enableLtsLogs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = EnableLtsLogsRequest()
        response = client.enable_lts_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.EnableLtsLogsRequest{}
response, err := client.EnableLtsLogs(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LTS日志成功。
400	请求错误
401	鉴权失败
403	没有操作权限
404	找不到资源
500	服务内部错误
503	服务不可用

5.16.3 关闭云服务日志

功能介绍

该接口用于关闭集群LTS云日志服务。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/lts-logs/disable

表 5-555 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 。
cluster_id	是	String	集群的ID。获取方法，请参见 获取集群ID 。

请求参数

无

响应参数

无

请求示例

```
https://{Endpoint}/v2/05f2cff45100d5112f4bc00b794ea08e/clusters/a862e17c-2981-4150-9018-f9a40511999c/lts-logs/disable
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class DisableLtsLogsSolution {

    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);
```

```
DwsClient client = DwsClient.newBuilder()
    .withCredential(auth)
    .withRegion(DwsRegion.valueOf("<YOUR REGION>"))
    .build();
DisableLtsLogsRequest request = new DisableLtsLogsRequest();
try {
    DisableLtsLogsResponse response = client.disableLtsLogs(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DisableLtsLogsRequest()
        response = client.disable_lts_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableLtsLogsRequest{}
    response, err := client.DisableLtsLogs(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	关闭LTS日志成功。
400	请求错误
401	鉴权失败
403	没有操作权限
404	找不到资源
500	服务内部错误
503	服务不可用

5.17 升级管理

5.17.1 获取集群升级记录

功能介绍

该接口用于获取当前集群升级记录。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/clusters/{cluster_id}/upgrade-management/records

表 5-556 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-557 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量。
limit	否	Integer	条目数。

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
count	Integer	集群升级记录总数。
records	Array of ClusterUpdateRecordResp objects	集群升级记录列表。

表 5-559 ClusterUpdateRecordResp

参数	参数类型	描述
item_id	String	升级项目ID。
status	String	升级状态： <ul style="list-style-type: none"> • Waiting：待升级。 • Update_Running：升级中。 • Update_Success：升级成功，待提交。 • Update_Failure：升级失败。 • Rollback_Running：回滚中。 • Rollback_Failure：回滚失败。 • Commit_Running：提交中。 • Commit_Failure：提交失败。 • Completed：升级完成。
record_type	String	升级类型。
from_version	String	升级前版本。
to_version	String	目标版本。
start_time	String	开始时间。
end_time	String	结束时间。
job_id	String	升级任务ID。
failed_reason	String	失败原因。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/c719b1a7-c85c-4cb5-a721-7694908c2c11/upgrade-management/records
```

响应示例

状态码： 200

集群升级记录响应体：

```
{
  "records": [
    {
      "item_id": "930d2820-fc17-4a42-9a50-eb695bf05e83",
      "status": "UpdateCompleted_Success",
      "record_type": "update-kernel",
      "from_version": "8.2.0",
      "to_version": "8.2.0.100",
      "start_time": "2023-08-04T06:46:46",
      "end_time": "2023-08-04T06:54:59",
      "job_id": "2c90804789bf441e0189bf4c78fd0001",
      "failed_reason": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListUpdateRecordSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListUpdateRecordRequest request = new ListUpdateRecordRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ListUpdateRecordResponse response = client.listUpdateRecord(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListUpdateRecordRequest()
        request.offset = <offset>
        request.limit = <limit>
        response = client.list_update_record(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ListUpdateRecordRequest{}
    offsetRequest := int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest := int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ListUpdateRecord(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	集群升级记录响应体。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.17.2 获取集群可升级的目标版本

功能介绍

该接口用于获取集群可升级的目标版本。

调用方法

请参见[如何调用API](#)。

URI

```
GET /v1/{project_id}/clusters/{cluster_id}/upgrade-management/avail-versions
```

表 5-560 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

表 5-561 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量。
limit	否	Integer	条目数。

请求参数

无

响应参数

状态码： 200

表 5-562 响应 Body 参数

参数	参数类型	描述
count	Integer	集群升级路径总条数。
items	Array of UpdateItemResp objects	集群升级路径列表。

表 5-563 UpdateItemResp

参数	参数类型	描述
id	String	升级项ID。
from	String	起始版本。
to	String	目标版本。
status	String	升级路径状态。
process	String	升级进度。
start_time	String	起始时间。
end_time	String	结束时间。
job_id	String	升级任务ID。
failed_reason	String	失败原因。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/c719b1a7-c85c-4cb5-a721-7694908c2c11/upgrade-management/avail-versions
```

响应示例

状态码： 200

集群升级路径响应体

```
{
  "items": [
    {
      "id": "930d2820-fc17-4a42-9a50-eb695bf05e83",
      "from": "8.2.0",
      "to": "8.2.0.100",
      "status": "Update_Success",
      "process": "0",
      "start_time": "2023-08-04T06:54:52",
      "end_time": "",
      "job_id": "2c90804789bf441e0189bf4c78fd0001",
      "failed_reason": 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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ListUpdatableVersionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ListUpdatableVersionRequest request = new ListUpdatableVersionRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ListUpdatableVersionResponse response = client.listUpdatableVersion(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ListUpdatableVersionRequest()
        request.offset = <offset>
        request.limit = <limit>
        response = client.list_updatable_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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
    dws.DwsClientBuilder().
        WithRegion(region.ValueOf("cn-north-4")).
        WithCredential(auth).
        Build())

request := &model.ListUpdatableVersionRequest{}
offsetRequest := int32(<offset>)
request.Offset = &offsetRequest
limitRequest := int32(<limit>)
request.Limit = &limitRequest
response, err := client.ListUpdatableVersion(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	集群升级路径响应体
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

5.17.3 下发集群升级相关操作

功能介绍

该接口用于下发集群升级相关操作。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/clusters/{cluster_id}/upgrade-management/action

表 5-564 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 。
cluster_id	是	String	集群ID。获取方法，请参见 获取集群ID 。

请求参数

表 5-565 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	当前集群要做的操作： <ul style="list-style-type: none"> • update • retry • rollback • commit
item_id	是	String	升级项ID。

响应参数

表 5-566 响应 Body 参数

参数	参数类型	描述
job_id	String	下发升级相关操作的任务ID。

请求示例

```
GET https://{Endpoint}/v1/0536cdee2200d5912f7cc00b877980f1/clusters/c719b1a7-c85c-4cb5-a721-7694908c2c11/upgrade-management/action
```

```
{
  "item_id": "xx",
  "action": "update"
}
```

响应示例

状态码： 200

```
{
  "job_id": "2c90807d89a6f9540189a9bef01f042b"
}
```

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.dws.v2.region.DwsRegion;
import com.huaweicloud.sdk.dws.v2.*;
import com.huaweicloud.sdk.dws.v2.model.*;

public class ExecuteClusterUpgradeActionSolution {

    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);

        DwsClient client = DwsClient.newBuilder()
            .withCredential(auth)
            .withRegion(DwsRegion.valueOf("cn-north-4"))
            .build();
        ExecuteClusterUpgradeActionRequest request = new ExecuteClusterUpgradeActionRequest();
        ExecuteClusterUpgradeActionRequestBody body = new ExecuteClusterUpgradeActionRequestBody();
        body.setItemId("xx");
        body.setAction("update");
        request.withBody(body);
        try {
            ExecuteClusterUpgradeActionResponse response = client.executeClusterUpgradeAction(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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdws.v2.region.dws_region import DwsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdws.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")

    credentials = BasicCredentials(ak, sk) \

    client = DwsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DwsRegion.value_of("cn-north-4")) \
        .build()

    try:
        request = ExecuteClusterUpgradeActionRequest(
            request.body = ExecuteClusterUpgradeActionRequestBody(
                item_id="xx",
                action="update"
            )
        )
        response = client.execute_cluster_upgrade_action(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"
    dws "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dws/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 := dws.NewDwsClient(
        dws.DwsClientBuilder().
            WithRegion(region.ValueOf("cn-north-4")).
            WithCredential(auth).
            Build())

    request := &model.ExecuteClusterUpgradeActionRequest{
        request.Body = &model.ExecuteClusterUpgradeActionRequestBody{
            ItemId: "xx",
            Action: "update",
        }
    }
    response, err := client.ExecuteClusterUpgradeAction(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	下发集群升级相关操作成功。
400	请求错误。
401	鉴权失败。
403	没有操作权限。
500	服务内部错误。
503	服务不可用。

6 应用示例

6.1 使用 Postman 调用创建集群接口

场景介绍

本章节指导用户使用Postman创建GaussDB(DWS)集群。该接口为异步接口，创建集群需要10~15分钟。API的调用方法请参考[如何调用API](#)。

准备工作

- 已注册华为云账号并开通华为云，完成实名认证，请参见[注册华为账号并开通华为云](#)。
- 已获取服务对应区域终端节点地址，具体请参考[地区和终端节点](#)。
- 已获取华为云账号对应的用户名、用户密码、账号名及项目ID。在华为云控制台右上角“我的凭证->API凭证->项目列表”查看用户名、账号名、用户密码和所属区域项目ID。项目ID获取方法请参见[获取项目ID](#)。
- 创建VPC，并获取VPC和子网的ID，具体参考[虚拟私有云VPC](#)。

操作步骤

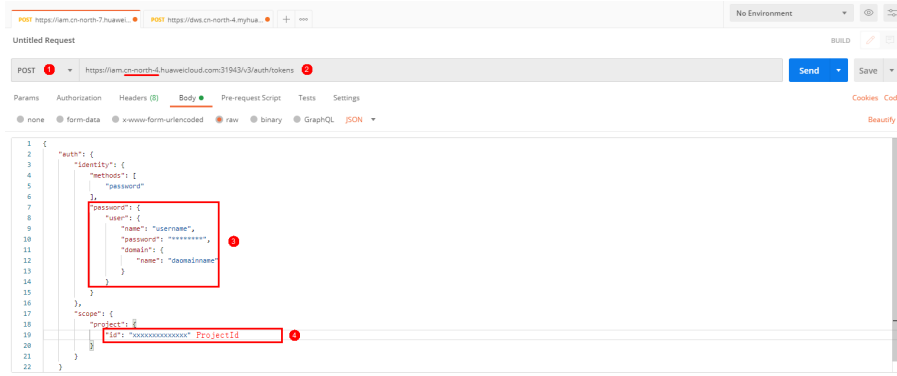
步骤1 环境配置。

说明

下载Postman调用创建快照服务，Postman建议使用7.24.0版本。

步骤2 获取Token，进行认证。

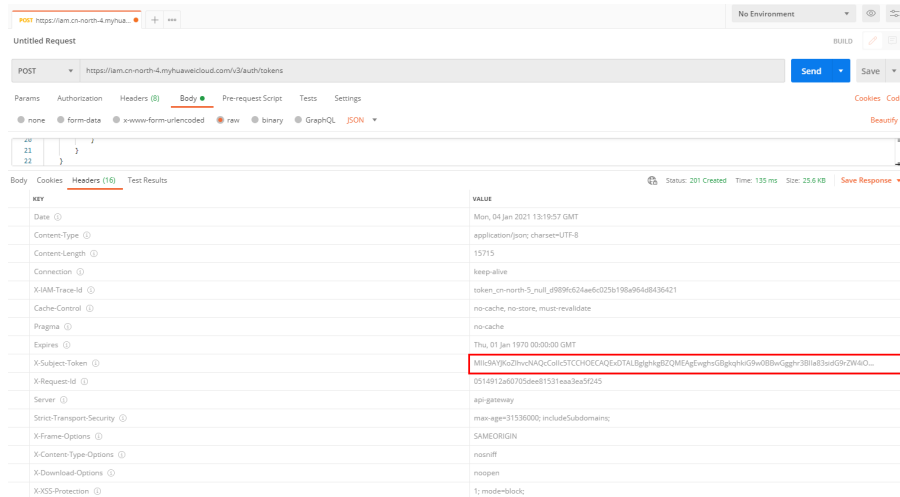
1. 在Postman界面上新建Post请求，选中Body的配置项，填写用户名、用户密码、账号名及项目ID。



说明

获取token的华为云服务所在区域需要和调用的服务所在区域需一致，否则会导致调用失败。

2. 单击右上角“Send”，发送“POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens”。获取返回结果中的Headers->X-Subject-Token值（Token有效期为24小时）。



步骤3 调用创建集群接口。

1. 在Postman界面上新建Post请求，选中“Headers”配置项，添加对应的KEY、VALUE，复制Token值到“X-Auth-Token”。
2. 调用接口，本服务提供了两个创建集群接口。其中请求URL的所在区域需要与调用的服务所在区域保持一致，project_id为项目ID（必填参数），获取方法参见[准备工作](#)。

- v1接口：
POST https://{Endpoint}/v1.0/{project_id}/clusters
- v2接口：
POST https://{Endpoint}/v2/{project_id}/clusters

说明

推荐使用v2接口。

3. 单击“Body”，添加请求参数值到请求中。不同服务的参数各不相同，详情请参考[API说明](#)。

```
{
  "cluster": {
```

```
"flavor": "dwsx2.rt.xlarge.m6",
"num_node": 3,
"subnet_id": "050e154d-9954-4929-b6d3-12af7c9213ac",
"security_group_id": "",
"vpc_id": "050e154d-9954-4929-b6d3-12af7c9213ac",
"db_port": 8000,
"datastore_version": "8.1.3.320",
"availability_zones": [
  "cn-north-4c"
],
"name": "dws1",
"num_cn": 3,
"db_name": "dbadmin",
"db_password": "Passw0rd!",
"public_ip": {
  "public_bind_type": "auto_assign",
  "eip_id": ""
},
"volume": {
  "volume": "SSD",
  "capacity": "100"
}
}
```

4. 单击右上角“Send”按钮发送请求。例如：“POST https://dws.cn-north-4.myhuaweicloud.com/v2/89cd04f168b84af6be287f71730fdb4b/clusters”，查看创建结果。

```
{
  "cluster": {
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15"
  }
}
```

----结束

常见问题

- DWS.5207 CN实例个数不合法
根据最新版本，集群CN数量，取值范围为2~集群节点数，最大值为20，默认值为3。
- DWS.5240 所选规格不存在
v2接口flavor填写规格名称，请勿使用规格ID。
- ECS.0319 创建集群失败，报错“The capacity of flavor am7.xlarge.8 is not sufficient.”。
所选DWS规格在底层资源不足，请确认flavor字段并重新选择可用的DWS规格。

6.2 使用 Postman 调用创建快照接口

场景描述

本章节指导用户使用Postman调用创建快照接口。API的调用方法请参考[如何调用API](#)。

涉及接口

调用该接口时，需要进行认证鉴权，涉及的接口如下：

- **认证鉴权**：经过身份认证，获得操作API的权限。
- **创建快照**：通过该接口创建集群的快照。

操作步骤

步骤1 配置环境。

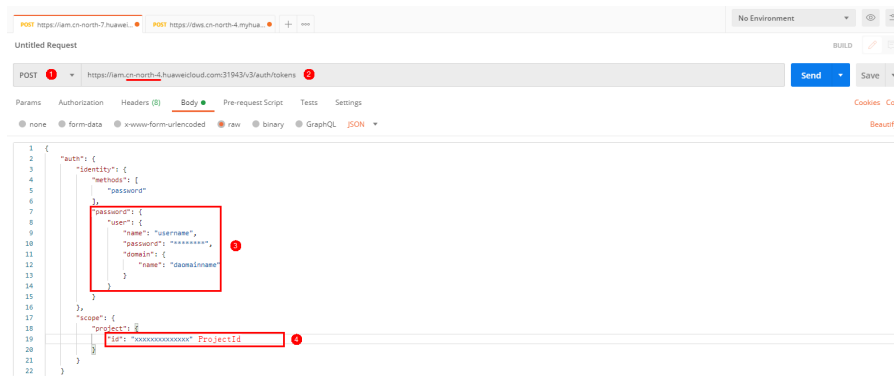
📖 说明

下载Postman调用创建快照服务，Postman建议使用7.24.0版本。

步骤2 调用服务。

1. 获取Token，进行认证。

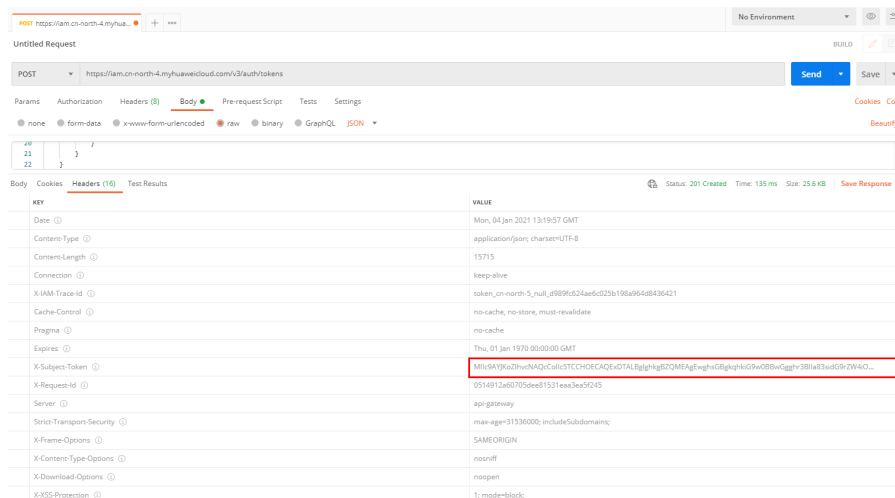
在Postman界面上新建Post请求，选中Body的配置项，填写用户名、用户密码、账号名及项目ID。在[我的凭证](#)下，您可以查看账号的用户名、用户密码和账号名等信息。项目ID获取方法请参见[获取项目ID](#)。



📖 说明

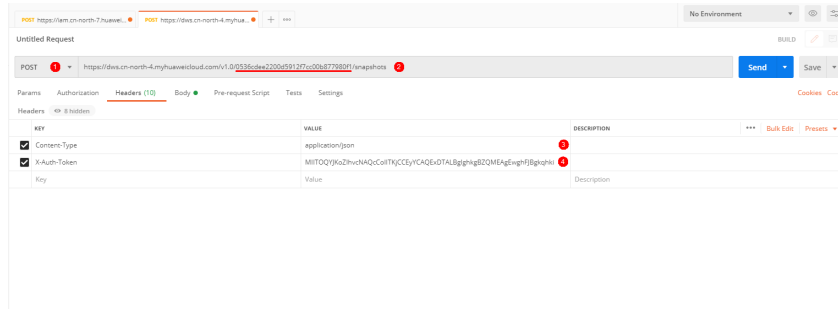
获取token的华为云服务所在区域需要和调用的服务所在区域需一致，否则会导致调用失败。

2. 单击右上角“Send”，发送“POST https://iam.cn-north-4.myhuaweicloud.com/v3/auth/tokens”。获取返回结果中的Token值（Token有效期为24小时）。

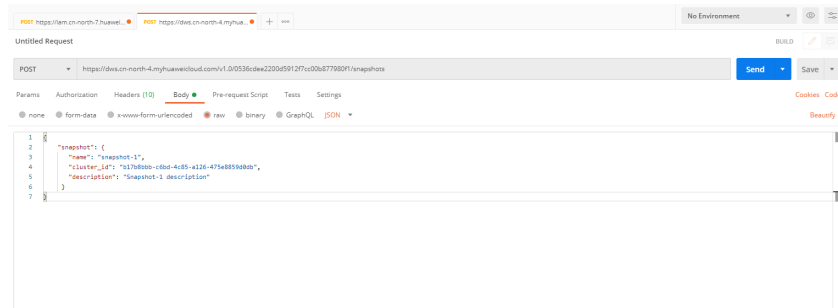


3. 调用创建快照服务。

- 在Postman界面上新建Post请求，选中“Headers”配置项，添加对应的KEY、VALUE，复制Token值到“X-Auth-Token”。请求中划红横线部分为项目ID，获取方法请参见[获取项目ID](#)。请求URL中的所在区域需要与调用的服务所在区域保持一致，您可以从[地区和终端节点](#)获取。



- 单击“Body”，添加请求参数值到请求中。不同服务的参数各不相同，详情请参考[API说明](#)。



- 单击右上角“Send”按钮发送请求“POST https://dws.cn-north-4.myhuaweicloud.com/v1.0/{project_id}/snapshots”，查看创建结果。

```
{
  "snapshot": {
    "id": "809bcf6e-4022-496d-9320-bbc700695692"
  }
}
```

----结束

7 权限及授权项说明

如果您需要对您所拥有的GaussDB(DWS)服务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），如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用GaussDB(DWS)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。用户通过创建自定义策略进行授权，具体操作请参见[GaussDB\(DWS\)自定义策略](#)。

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GaussDB(DWS)集群列表，那么这个IAM用户被授予的策略中必须包含允许“dws:openAPICluster: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**：允许或拒绝对指定资源在特定条件下进行某项操作。
- **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。
- **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。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](#)。

 说明

“√”表示支持，“x”表示暂不支持。

GaussDB(DWS) 支持的自定义策略授权项如下所示

- [集群管理](#)
- [快照管理](#)

集群管理

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建集群	POST /v1.0/{project_id}/clusters	dws:openAPICluster:create	√	√
查询集群列表	GET /v1.0/{project_id}/clusters	dws:openAPICluster:list	√	√
查询集群详情	GET /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:getDetail	√	√
查询节点类型	GET /v2/{project_id}/node-types	dws:openAPIFlavors:get	√	√
删除集群	DELETE /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:delete	√	√
重启集群	POST /v1.0/{project_id}/clusters/{cluster_id}/restart	dws:openAPICluster:restart	√	√
扩容集群大小	POST /v1.0/{project_id}/clusters/{cluster_id}/resize	dws:cluster:scaleOutOrOpenAPIResize	√	√

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
重置集群管理员密码	POST /v1.0/{project_id}/clusters/{cluster_id}/reset-password	dws:openAPICluster:resetPassword	√	√
启动集群	POST /v1/{project_id}/clusters/{cluster_id}/start	dws:cluster:startCluster	√	×
停止集群	POST /v1/{project_id}/clusters/{cluster_id}/stop	dws:cluster:stopCluster	√	×

📖 说明

当前版本启动集群、停止集群接口仅对数据库类型为serverless的集群开放。

快照管理

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
创建快照	POST /v1.0/{project_id}/snapshots	dws:openAPISnapshot:create	√	√
查询快照列表	GET /v1.0/{project_id}/snapshots	dws:openAPISnapshot:list	√	√
查询快照详情	GET /v1.0/{project_id}/snapshots/{snapshot_id}	dws:openAPISnapshot:detail	√	√
删除快照	DELETE /v1.0/{project_id}/snapshots/{snapshot_id}	dws:openAPISnapshot:delete	√	√

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
恢复集群	POST /v1.0/ {project_id}/ snapshots/ {snapshot_id}/ actions	dws:openAPISna pshot:restore	√	√

8 附录

8.1 状态码

状态码如表8-1所示

表 8-1 状态码

状态码	编码	状态说明
100	Continue	继续请求。 该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	状态说明
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。

状态码	编码	状态说明
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。
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。

8.2 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息

体中是具体的错误代码及错误信息。在调用方找不到错误原因时，可以联系数据库客服和技术支持，并提供错误码，以便尽快帮您解决问题。

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。

表 8-2 错误码说明

状态码	错误码	错误信息	描述	处理措施
400	DWS.5001	实例名非法。	实例名非法。	请参考对应实例名的参数说明输入符合要求的实例名后重试。
400	DWS.5002	数据库类型非法。	数据库类型非法。	请参考对应数据库的参数说明输入符合要求的数据库类型后重试。
400	DWS.5003	数据库版本非法。	数据库版本非法。	请参考对应数据库的参数说明输入符合要求的数据库版本后重试。
400	DWS.5004	datastore字段为空。	datastore字段为空。	请根据实际情况输入正确的datastore后重试。
400	DWS.5005	不支持的数据库类型或版本。	不支持的数据库类型或版本。	请参考对应数据库的参数说明输入符合要求的数据库类型和版本后重试。
400	DWS.5006	无效的规格。	无效的规格。	请输入正确的规格后重试。
400	DWS.5010	无效区域。	无效区域。	请重新选择有效的区域后重试。
400	DWS.5011	无效可用区。	无效可用区。	请重新选择有效的可用区后重试。
400	DWS.5012	区域或可用区不存在。	区域或可用区不存在。	请输入正确的区域或可用区后重试。
400	DWS.5013	数据库root密码非法。	数据库root密码非法。	请参考对应的密码参数说明，输入符合密码复杂度要求的密码。
400	DWS.5014	VPC id非法。	VPC id非法。	请参考vpc_id的参数说明，输入符合要求的id。
400	DWS.5015	子网id非法。	子网id非法。	请参考subnet_id的参数说明，输入符合要求的id。
400	DWS.5016	安全组id非法。	安全组id非法。	请参考安全组id的参数说明，输入符合要求的id。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5 017	自动备份保留天数无效。	自动备份保留天数无效。	请输入符合要求的自动备份保留天数后重试。
400	DWS.5 018	自动备份周期无效。	自动备份周期无效。	请输入符合要求的自动备份周期后重试。
404	DWS.5 021	VPC不存在, 或不属于该用户。	VPC不存在, 或不属于该用户。	请输入正确的VPC后重试。
404	DWS.5 022	子网不存在, 或不属于该VPC。	子网不存在, 或不属于该VPC。	请输入正确的子网后重试。
404	DWS.5 023	安全组不存在, 或不属于该VPC。	安全组不存在, 或不属于该VPC。	请输入正确的安全组后重试。
400	DWS.5 027	无效的Availability Zone。	无效的Availability Zone。	请输入正确的Availability Zone后重试。
400	DWS.5 033	volume的大小非10的倍数。	volume的size非10的倍数。	请输入正确的size后重试。
400	DWS.5 034	规格不匹配。	规格不匹配。	请选择合适的规格后重试。
403	DWS.5 036	Datastore不存在或没有权限。	Datastore不存在或没有权限。	请输入正确的Datastore后重试。
400	DWS.5 037	参数不存在。	参数不存在。	请输入正确的参数后重试
400	DWS.5 038	参数值超出范围。	参数值超出范围。	请输入正确的参数后重试。
400	DWS.5 039	参数重复。	参数重复。	请输入正确的参数后重试。
400	DWS.5 045	当前租户不允许操作!	当前租户不允许操作!	请确保当前租户与集群所属租户一致后重试。
400	DWS.5 046	nics为空。	nics为空。	请输入正确的nics后重试。
400	DWS.5 047	实例个数不合法!	实例个数不合法!	请输入合法的实例个数。
400	DWS.5 048	扩展参数不合法!	扩展参数不合法!	请参考对应的扩展参数说明, 输入符合要求的扩展参数。
409	DWS.5 050	集群名已经存在。	集群名已经存在。	请确保输入正确的集群名后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5051	ip非法。	ip非法。	请输入正确的ip后重试。
400	DWS.5052	可用区取值非法。	可用区取值非法。	请输入正确的可用区后重试。
400	DWS.5053	实例之间的亲和性取值非法。	实例之间的亲和性取值非法。	请输入正确的亲和性值后重试。
400	DWS.5054	实例参数组的id取值非法。	实例参数组的id取值非法。	请输入正确的实例参数组id后重试。
400	DWS.5056	String类型长度违规。	String类型长度违规。	请输入符合要求的String类型后重试。
400	DWS.5057	集群名称含非法字符或长度不合法。	集群名称含非法字符或长度不合法。	请参考集群名的参数说明，输入符合要求的集群名称。
400	DWS.5059	数据库用户名非法。	数据库用户名非法。	请参考数据库用户名的参数说明，输入符合要求的数据库用户名。
400	DWS.5060	Integer类型长度违规。	Integer类型长度违规。	请输入符合要求的Integer类型后重试。
400	DWS.5061	Volume类型违规。	Volume类型违规。	请输入符合要求的Volume类型后重试。
400	DWS.5062	Volume标签类型违规。	Volume标签类型违规。	请输入符合要求的Volume标签类型后重试。
400	DWS.5063	Specific字段无效。	Specific字段无效。	请输入正确的Specific值后重试。
400	DWS.5064	UUID非法。	UUID非法。	请输入正确的UUID后重试。
400	DWS.5065	格式非法。	格式非法。	请输入正确的格式后重试。
400	DWS.5070	规格信息与xml配置不匹配。	规格信息与xml配置不匹配。	请输入正确的规格信息后重试。
400	DWS.5071	磁盘类型与xml配置不匹配。	磁盘类型与xml配置不匹配。	请输入正确的磁盘类型后重试。
400	DWS.5078	磁盘大小不合法。	磁盘大小不合法。	请输入正确的磁盘大小后重试。
400	DWS.5079	备份周期不在允许范围内。	备份周期不在允许范围内。	请输入符合要求的备份周期后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5080	备份保留天数不在允许范围内。	备份保留天数不在允许范围内。	请输入符合要求的备份保留天数后重试。
400	DWS.5081	实例硬盘类型不在允许范围内。	实例硬盘类型不在允许范围内。	请选择符合要求的实例硬盘类型后重试。
400	DWS.5082	不支持的数据库用户名。	不支持的数据库用户名。	请参考数据库用户名的参数说明，输入符合要求的数据库用户名。
400	DWS.5083	亲和性非xml配置里允许的范围。	亲和性非xml配置里允许的范围。	请输入符合要求的亲和性值后重试。
400	DWS.5084	XML文件中Volume配置有误。	XML文件中Volume配置有误。	请配置正确的Volume值后重试。
400	DWS.5086	同一实例类型不满足规格一致性。	同一实例类型不满足规格一致性。	请输入正确的规格后重试。
400	DWS.5087	同一实例类型不满足磁盘一致性。	同一实例类型不满足磁盘一致性。	请输入正确的磁盘后重试。
400	DWS.5089	不支持创建的类型。	不支持创建的类型。	请输入正确的类型后重试。
400	DWS.5091	节点个数达到最大值。	节点个数达到最大值。	请重新输入正确的节点个数后重试。
400	DWS.5093	实例规格信息不合法。	实例规格信息不合法。	请输入符合要求的实例规格后重试。
400	DWS.5095	当前实例类型不支持此操作。	当前实例类型不支持此操作。	请选择正确的实例类型后重试。
400	DWS.5096	磁盘个数不合法。	磁盘个数不合法。	请输入符合要求的磁盘个数后重试。
400	DWS.5097	Region或Availability Zone不存在。	Region或Availability Zone不存在。	请输入正确的Region和Availability Zone后重试。
400	DWS.5098	EIP绑定类型不合法。	EIP绑定类型不合法。	请输入正确的EIP绑定类型后重试。
400	DWS.5104	实例的子网必须唯一。	实例的子网必须唯一。	请确认实例的子网唯一后重试。
400	DWS.5106	实例类型非法!	实例类型非法!	请输入符合要求的实例类型后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5 107	集群模式非法!	集群模式非法!	请输入符合要求的集群模式后重试。
400	DWS.5 111	备份文件id不存在!	备份文件id不存在!	请输入正确的备份文件id后重试。
400	DWS.5 130	委托名称非法。	委托名称非法。	请输入正确的委托名称后重试。
400	DWS.5 133	EIP id非法。	EIP id非法。	请输入正确的EIP id后重试。
400	DWS.5 135	EIP不存在。	EIP不存在。	请输入正确的EIP后重试。
400	DWS.5 144	集群管理员密码非法。	集群管理员密码非法。	请参考集群管理员密码参数说明，输入符合要求的集群管理员密码。
400	DWS.5 145	集群管理员密码不能与用户名或倒序的用户名相同。	集群管理员密码不能与用户名或倒序的用户名相同。	请根据错误信息提示输入符合要求的集群管理员密码后重试。
400	DWS.5 146	云服务器组参数缺失。	云服务器组参数缺失。	请输入正确的参数后重试。
400	DWS.5 147	云服务器组参数不合法。	云服务器组参数不合法。	请输入正确的参数后重试。
400	DWS.5 148	云服务器组个数达到最大值。	云服务器组个数达到最大值。	请输入正确的云服务器组个数后重试。
400	DWS.5 149	快照id不合法。	快照id不合法。	请参考快照id的参数说明，输入符合要求的快照id。
400	DWS.5 165	数据文件中空值内容不合法。	数据文件中空值内容不合法。	请输入正确的控制内容后重试。
400	DWS.5 166	TEXT格式下的noescaping参数不合法。	TEXT格式下的noescaping参数不合法。	请输入正确的noescaping参数内容后重试。
400	DWS.5 167	数据文件的header参数不合法，支持范围为true/on, false/off。	数据文件的header参数不合法，支持范围为true/on, false/off。	请输入正确的header参数内容后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5 168	数据文件的 fill_missing_fields 参数不合法, 支持范围为 true/on, false/off。	数据文件的 fill_missing_fields 参数不合法, 支持的取值为 true/on, false/off。	请输入正确的 fill_missing_fields 参数内容后重试。
400	DWS.5 169	数据文件的 ignore_extra_data 参数不合法, 支持范围为 true/on, false/off。	数据文件的 ignore_extra_data 参数不合法, 支持的取值为 true/on, false/off。	请输入正确的 ignore_extra_data 参数内容后重试。
400	DWS.5 170	数据导入允许出现的数据格式错误个数输入不合法。	数据导入允许出现的数据格式错误个数输入不合法。	请输入正确的错误个数后重试。
400	DWS.5 171	数据库登录用户名或密码不合法。	数据库登录用户名或密码不合法。	请输入正确的用户名或密码后重试。
400	DWS.5 172	数据库不存在。	数据库不存在。	请输入正确的数据库名后重试。
400	DWS.5 173	数据库表所在 schema 不存在。	数据库表所在 schema 不存在。	请输入正确的 schema 后重试。
400	DWS.5 174	数据库表不存在。	数据库表不存在。	请输入正确的数据库表后重试。
400	DWS.5 175	数据库用户名不能为空。	数据库用户名不能为空。	请输入用户名后重试。
400	DWS.5 176	数据库用户名对应密码不能为空。	数据库用户名对应密码不能为空。	请输入密码后重试。
400	DWS.5 177	指定加载数据的表所在数据库名不能为空。	指定加载数据的表所在数据库名不能为空。	请输入数据库名后重试。
400	DWS.5 178	指定加载数据的表所在 schema 不能为空。	指定加载数据的表所在 schema 不能为空。	请输入 schema 后重试。
400	DWS.5 179	指定加载数据的数据库表不能为空。	指定加载数据的数据库表不能为空。	请输入数据库表后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5 180	OBS加载文件路径不合法。	OBS加载文件路径不合法。	请输入正确的OBS加载路径后重试。
400	DWS.5 181	数据加载失败， 错误信息： #failedReason# 。	数据加载失败， 错误信息： #failedReason# 。	请根据错误提示进行修复后重试。
400	DWS.5 182	当前任务已结束	当前任务已结束	请选择正在加载的任务进行取消操作。
400	DWS.5 183	当前任务已被取消	当前任务已被取消	请选择正在加载的任务进行取消操作。
400	DWS.5 184	数据源文件的格式不合法，支持格式为CSV、TEXT。	数据源文件的格式不合法，支持格式为CSV、TEXT。	请输入正确的数据源文件格式后重试，支持格式为CSV、TEXT。
400	DWS.5 185	数据文件的编码格式不合法，支持格式为GBK、UTF8、Latin1、SQL_ASCII。	数据文件的编码格式不合法，支持格式为GBK、UTF8、Latin1、SQL_ASCII。	请输入正确的编码格式后重试，支持格式为GBK、UTF8、Latin1、SQL_ASCII。
400	DWS.5 186	数据文件行数据的字段分隔符不合法。	数据文件行数据的字段分隔符不合法。	请输入正确的字段分隔符后重试。
400	DWS.5 187	CSV格式文件下的引号字符不合法。	CSV格式文件下的引号字符不合法。	请输入正确的引号字符后重试。
400	DWS.5 189	获取ak/sk失败。	获取ak/sk失败。	请联系数据库客服或者稍后重试。
400	DWS.5 191	标签不存在。	标签不存在。	请输入正确的标签后重试。
400	DWS.5 194	标签个数超过最大值。	标签个数超过最大值。	请检查资源上的标签个数后重试。
400	DWS.5 197	标签非法。	标签非法。	请输入正确的标签后重试。
400	DWS.5 198	用户名不合法。	用户名不合法。	请输入正确的用户名后重试。
400	DWS.5 199	密码不合法。	密码不合法。	请输入正确的密码后重试。

状态码	错误码	错误信息	描述	处理措施
400	DWS.5200	当前规格已售罄，请切换其他可用区或区域。	当前规格已售罄，请切换其他可用区或区域。	请创建其他规格集群或切换其他可用区或区域。
400	DWS.5202	子网ID不能为空。	子网ID不能为空。	请输入正确的子网ID后重试。
400	DWS.5203	安全组ID不能为空。	安全组ID不能为空。	请输入正确的安全组ID后重试。
400	DWS.5206	数据库版本不支持该功能。	数据库版本不支持该功能。	请确保接口参数与集群版本匹配，若不匹配，请将参数修改正确后再重试。

8.3 创建 VPC

背景信息

在创建集群之前，您需要创建VPC，为GaussDB(DWS) 服务提供一个安全、隔离的网络环境。

如果用户已有VPC，可重复使用，不需多次创建。

📖 说明

关于创建VPC的详细信息可以参考《虚拟私有云用户指南》中的[创建虚拟私有云和子网](#)章节。

操作步骤

步骤1 登录管理控制台。

步骤2 选择“网络 > 虚拟私有云”。

步骤3 在“虚拟私有云”界面，单击“创建虚拟私有云”，创建虚拟私有云。

步骤4 获取VPC及子网id，后续[创建集群](#)时需要使用。

步骤5 在“虚拟私有云”界面，在左侧导航树，选择“访问控制 > 安全组”，再单击“创建安全组”，创建安全组。

步骤6 获取安全组id，后续[创建集群](#)时需要使用。

----结束

8.4 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。

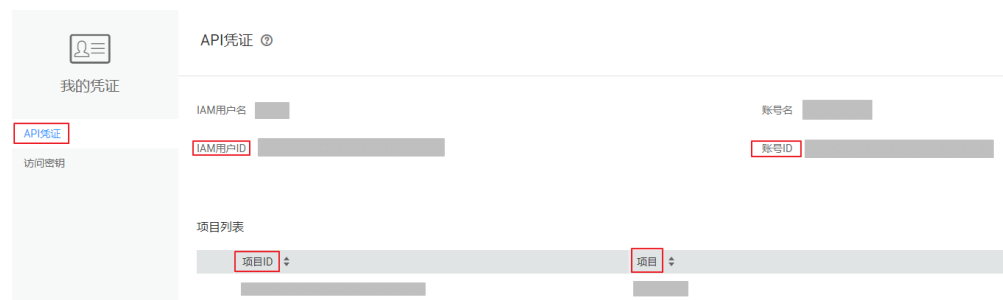
```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "cn-north-4b",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "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中需要填入项目编号，所以需要获取到项目编号。项目编号获取步骤如下：

1. 登录管理控制台。
2. 单击用户名，在下拉列表中单击“基本信息”。
3. 在基本信息页面单击“管理我的凭证”。
在“API凭证”页面的项目列表中查看项目ID。

图 8-1 查看项目 ID



8.5 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id），所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。

在“API凭证”页面中查看账号ID。

图 8-2 获取账号 ID



8.6 获取集群 ID

在调用接口的时候，部分URL中需要填入集群ID (cluster_id)，所以需要获取到集群ID。获取方法如下所示：

调用 API 获取集群 ID

集群ID可以通过调用[查询集群列表](#)API接口获取。

获取集群ID的接口为“GET https://{Endpoint}/v1.0/{project_id}/clusters”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取；项目ID参见[获取项目ID](#)；接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，在该项目找到指定clusters下的“id”即为集群ID。

```
{
  "clusters": [ {
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
    "name": "dws-1",
    "status": "AVAILABLE",
    "version": "1.2.0",
    "updated": "2016-02-10T14:28:14Z",
    "created": "2016-02-10T14:26:14Z",
    "port": 8000,
    "endpoints": [ {
      "connect_info": "dws-1.cn-north-4.myhuaweicloud.com",
      "jdbc_url": "jdbc:postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
    } ],
    "nodes": [ {
      "id": "acaf62a4-41b3-4106-bf6b-2f669d88291e",
      "status": "200"
    }, {
      "id": "d32de51e-4fcd-4e5a-a9dc-bb903abb494b",
      "status": "200"
    }, {
      "id": "d71a4a25-c9bc-4ffd-9f4a-e422aef327f9",
      "status": "200"
    } ],
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value2"
    } ],
    "user_name": "dbadmin",
    "number_of_node": 3,
    "recent_event": 6,
    "availability_zone": "cn-north-4b",
    "enterprise_project_id": "0",
    "node_type": "dws.d1.xlarge.ultrahigh",
  } ]
}
```

```

"vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
"subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
"public_ip": {
  "public_bind_type": "auto_assign",
  "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
},
"public_endpoints": [ {
  "public_connect_info": "dws-1.cn-north-4.myhuaweicloud.com",
  "jdbc_url": "jdbc:postgresql://dws-1.cn-north-4.myhuaweicloud.com/<YOUR_DATABASE_name>"
} ],
"action_progress": {
  "SNAPSHOTTING": "20%"
},
"sub_status": "READONLY",
"task_status": "SNAPSHOTTING",
"security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b"
} ]
}
    
```

从控制台获取集群 ID

- 步骤1 登录GaussDB(DWS)管理控制台。
- 步骤2 在左侧导航栏，单击“集群管理”。
- 步骤3 在集群列表中找到所需要的集群，然后单击集群名称，进入集群“集群详情”页面。
- 步骤4 在“基本信息”模块查看集群ID，如下图所示。

图 8-3 查看集群 ID



----结束

8.7 自定义指标列表

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
AbnormalJob	异常作业监控统计	ctime	Timestamp	采集时间。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		virtual_cluster_id	Integer	集群虚拟ID。
		query_id	String	查询ID。
		statement	String	执行了异常处理的语句。
		block_time	Long	语句执行前的阻塞时间。
		elapsed_time	Long	语句的实际执行时间。
		total_cpu_time	Long	语句执行异常处理时DN上CPU使用的总时间。
		cpu_skew_percent	Integer	语句在执行异常处理时DN上CPU使用的倾斜率。
		control_group	String	语句执行异常处理时所使用的cgroups。
		status	String	语句执行异常处理后的状态。
		action	String	语句执行的异常处理动作。
abort_info	String	语句被异常处理的原因。		
AgentStatus	DMS集群Agent状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		is_ccn	Boolean	是否为CCN节点。
		agent_version	String	与DMS版本适配的Agent版本。
		agent_pid_exist	Boolean	pid文件是否存在。
		agent_start_time	String	Agent启动时间。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		agent_run_time	String	Agent运行时间。
		agent_sub_thread	Integer	Agent子线程。
		config_plugin_count	Integer	配置的plugin数量。
		active_plugin_count	Integer	启用的plugin数量。
		config_custom_plugin_count	Integer	配置的自定义plugin数量。
		active_custom_plugin_count	Integer	启用的自定义plugin数量。
		agent_cpu_consume	Float	AgentCPU消耗占比。
		agent_mem_consume	Float	Agentmem消耗占比。
		workplace_dir_time	String	workpalce目录更新时间。
		config_dir_time	String	config目录更新时间。
		log_file_time	String	log文件更新时间。
		perflog_file_time	String	perflog文件更新时间。
		timestamp_yaml_detail	String	TOPSQL时间戳。
		status_log_detail	String	状态记录详情。
ClusterHostStatus	节点状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		host_name	String	主机名。
		host_stat	String	主机状态。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
ClusterInstStatus	实例状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		host_name	String	主机名。
		inst_id	Integer	实例ID。
		inst_type	String	实例类型。
		inst_role	String	实例角色。
		inst_stat	String	实例状态。
		inst_stat_reason	String	实例状态原因。
ClusterSlowInst	慢实例统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		check_time	Long	检测时间。
		host_id	Integer	采集上报的节点ID。
		host_name	String	主机名。
		inst_id	String	实例ID。
		inst_name	String	实例名称。
ClusterStatus	集群状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		cluster_state	String	集群状态。
		balanced	String	是否发生过主备切换。
		redistributing	String	是否发生过重分布。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		cluster_read_only	String	当前集群是否只读。
CpuStat	CPU状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	String	采集上报的节点ID。
		cpu_name	String	默认情况下为ALL。
		usr	Float	用户态CPU时间%。
		nice	Float	nice值为负进程的CPU时间%。
		sys	Float	内核态时间%。
		iowait	Float	io等待时间%。
		irq	Float	硬中断时间%。
		soft	Float	软中断时间%。
		steal	Float	虚拟机管理器在服务另一个虚拟处理器时虚拟CPU处在非自愿等待下花费时间%。
		guest	Float	运行虚拟处理器花费的时间%。
		idle	Float	除去磁盘等待操作之外的空闲时间%。
		ht_possible	Boolean	是否有超线程能力。
		ht_enabled	Boolean	是否开启超线程。
run_count	Long	处于runnable状态的进程个数。		
blocked_count	Long	处于等待I/O完成状态的进程个数。		
DbActiveStats	活跃会话	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		db_name	String	数据库名称。
		inst_name	String	实例名称，为了说明当前值从哪个CN上取到的。
		total_sessions_count	Integer	所有用户会话数量。
		user_count	Integer	distinct用户名。
		application_count	Integer	distinct应用名称。
		active_queries	Integer	处于“active”、“fastpathfunctioncall”状态的查询数量。
DbSize	磁盘容量统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		inst_name	String	实例名称，为了说明当前值从哪个CN上取到的。
		db_name	String	数据库名称。
		db_size	Long	TSField。
DbStats	事务状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		db_name	String	数据库名称。
		inst_name	String	实例名称，为了说明当前值从哪个CN上取到的。
		tup_returned	Long	通过数据库全局扫描返回的行数。
		tup_fetched	Long	通过数据库查询索引返回的行数。
		tup_inserted	Long	通过数据库查询插入的行数。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		tup_updated	Long	通过数据库查询更新的行数。
		tup_deleted	Long	通过数据库查询删除的行数。
		xact_commit	Long	此数据库中已经提交的事务数。
		xact_rollback	Long	此数据库中已经回滚的事务数。
		deadlocks	Long	在该数据库中检索的死锁数。
		blks_read	Long	在这个数据库中读取的磁盘块的数量。
		blks_hit	Long	高速缓存中已经发现的磁盘块的次数，这样读取是不必要的（这只包括postgresql缓冲区高速缓存，没有操作系统的文件系统缓存）。
		blk_read_time	Double	通过数据库后端读取数据文件块花费的时间，以毫秒计算。
		blk_write_time	Double	通过数据库后端写入数据文件块花费的时间，以毫秒计算。
		conflicts	Long	由于数据库恢复冲突取消的查询数量。（只在备用服务器发生的冲突）。
		temp_files	Long	通过数据库查询创建的临时文件数量。计算所有临时文件（比如排序或者哈希），且不管log_temp_files设置。
		temp_bytes	Long	通过数据库查询写入临时文件的数据总量。计算所有临时文件，且不管log_temp_files设置
		db_size	Long	数据库容量（单位：字节）。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		delta_tup_returned	Long	单位时间内通过数据库全局扫描返回的行数。
		delta_tup_fetched	Long	单位时间内通过数据库查询索引返回的行数。
		delta_tup_inserted	Long	单位时间内通过数据库查询插入的行数。
		delta_tup_updated	Long	单位时间内通过数据库查询更新的行数。
		delta_tup_deleted	Long	单位时间内通过数据库查询删除的行数。
		delta_xact_commit	Long	单位时间内此数据库中已经提交的事务数。
		delta_xact_rollback	Long	单位时间内此数据库中已经回滚的事务数。
		delta_deadlocks	Long	单位时间内在该数据库中检索的死锁数。
FileSystemInodeStat	文件句柄	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		file_system_name	String	磁盘文件系统名称。
		total	Long	inode总容量（单位：KB）。
		used	Long	已使用容量情况（单位：KB）。
FileSystemStat	节点磁盘使用	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		disk_fs_name	String	磁盘文件系统名称。
		total	Long	总容量（单位：KB）。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		used	Long	已使用容量（单位：KB）。
		available	Long	可用容量（单位：KB）。
		used_percent age	Float	磁盘使用率%。
IOStat	节点磁盘I/O统计	ctime	Timest amp	采集时间。
		virtual_cluster _id	Intege r	集群虚拟ID。
		host_id	Intege r	采集上报的节点ID。
		disk_name	String	磁盘名称 （ devicename ）。
		tps	Float	每秒传输次数 （ transferpersecond ） 每次传输的大小未知。
		kb_read_s	Float	每秒从设备读取的数据 量（单位：KB）。
		kb_wrtn_s	Float	每秒向设备写入的数据 量（单位：KB）。
		kb_read	Float	读取的总数据量（单 位：KB）。
		kb_wrtn	Float	写入的总数据量（单 位：KB）。
		rrqm_s	Float	每秒对该设备的读请求 被合并的次数。
		wrqm_s	Float	每秒对该设备的写请求 被合并的次数。
		r_s	Float	每秒完成读次数。
		w_s	Float	每秒完成写次数。
		rkb_s	Float	每秒读数据量（单位： KB）。
wkb_s	Float	每秒写数据量（单位： KB）。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		avgrq_sz	Float	平均每次I/O操作的数据量（单位：扇区数）。
		avgqu_sz	Float	平均请求队列长度。
		await	Float	平均每次io请求的等待时间（单位：ms）。
		svctm	Float	平均每次io请求的处理时间（单位：ms）。
		util	Float	在统计时间内所有处理I/O时间，除以总共统计时间，即I/O队列非空的时间比例，百分比。
InstanceMemory	实例内存监控统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		inst_name	String	实例名称。
		max_process_memory	Float	实例所占用的内存大小。
		process_used_memory	Float	进程所使用的内存大小。
		max_dynamic_memory	Float	最大动态内存。
		dynamic_used_memory	Float	已使用的动态内存。
		dynamic_peak_memory	Float	内存的动态峰值。
		dynamic_used_shrctx	Float	最大动态共享内存上下文。
		dynamic_peak_shrctx	Float	共享内存上下文的动态峰值。
		max_shared_memory	Float	最大共享内存。
		shared_used_memory	Float	已使用的共享内存。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		max_cstore_memory	Float	列存所允许使用的最大内存。
		cstore_used_memory	Float	列存已使用的内存大小。
		max_comm_memory	Float	通信库所允许使用的最大内存。
		comm_used_memory	Float	通信库已使用的内存大小。
		comm_peak_memory	Float	通信库的内存峰值。
		max_topsql_memory	Float	TopSQL记录历史作业监控信息允许使用的最大内存。
		topsql_peak_memory	Float	TopSQL记录历史作业监控信息的内存峰值。
		topsql_used_memory	Float	TopSQL记录历史作业监控信息已使用的内存大小。
		other_used_memory	Float	其他已使用的内存大小。
		pooler_conn_memory	Float	pooler连接占用内存大小。
		pooler_freconn_memory	Float	pooler空闲连接占用的内存大小。
		storage_compress_memory	Float	列存压缩和解压缩使用的内存大小。
		udf_reserved_memory	Float	为UDFWorker进程预留的内存大小。
		mmap_used_memory	Float	mmap使用的内存大小。
InstanceResources	实例资源统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		inst_name	String	实例名称。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		inst_cpu	Float	读取 “postmaster.pID/cm_server.pID/gtm.pID/etcd.pID” 中的值（CPU 使用率%）。
		inst_mem	Float	读取 “postmaster.pID/cm_server.pID/gtm.pID/etcd.pID” 中的值（内存使用率%）。
InstanceSize	实例磁盘大小统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		inst_name	String	实例名称。
		inst_path	String	存储位置。
		inst_size	Long	当前实例上所有数据库使用的磁盘空间。
MemStat	节点内存统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		mem_total	Long	所有可用ram大小，物理内存减去预留位和内核使用量后的剩余值（单位：KB）。
		mem_free	Long	系统中未使用的内存，lowfree+highfree（单位：KB）。
		buffers	Long	用来给块设备做缓存的大小（单位：KB）。
		cached	Long	文件缓冲区大小（单位：KB）。
		swap_total	Long	交换空间总和（单位：KB）。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		swap_free	Long	ram暂存在swap中的大小（单位：KB）。
		hardware_corrupted	Long	中毒页面中的内存量（单位：KB）。
NetIfStat	网络状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		interface_name	String	网卡名称。
		up	Boolean	网卡状态（up/down）。
		speed	Long	网卡速度（千兆/百兆）。
		recv_bytes	Long	网卡接收总数据（单位：字节）。
		recv_packets	Long	网卡接收数据的总包数。
		recv_errors	Long	网卡接收错误总数。
		recv_drop	Long	网卡接收时丢弃的数据包总数。
		recv_fifo	Long	接收时fifo缓冲区错误的数量。
		recv_frame	Long	接收时分组帧错误的数量。
		recv_compressed	Long	接收到的压缩数据包数量。
		recv_multicast	Long	接收到的多播帧数量。
		send_bytes	Long	网卡发送总数据（单位：字节）。
send_packets	Long	网卡发送数据的总包数。		
send_errors	Long	网卡发送错误总数。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		send_drop	Long	网卡发送时丢弃的数据包总数。
		send_fifo	Long	发送时fifo缓冲区错误的数量。
		send_collisions	Long	发送接口上检测到的冲突数。
		send_carrier	Long	发送时由设备驱动程序检测到的载波损耗的数量。
		send_compressed	Long	发送出的压缩数据包数量。
		multi_q_possible	Boolean	是否支持网卡多队列。
		multi_q_enabled	Boolean	网卡多队列开启。
		multi_q_smp_affinity	String	多队列网卡的CPU亲和性。
		duplex	String	网卡是否双工。
		io_rate	Double	网络速率。
NodeSqlCount	节点sql统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		node_name	String	节点名。
		user_name	String	用户名。
		select_count	Long	SELECT数量。
		update_count	Long	UPDATE数量。
		insert_count	Long	INSERT数量。
		delete_count	Long	DELETE数量。
		mergeinto_count	Long	MERGEINTO数量。
		ddl_count	Long	DDL数量。
		dml_count	Long	DML数量。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		dcl_count	Long	DCL数量。
		total_select_elapsed	Long	SELECT总响应时间。
		avg_select_elapsed	Long	SELECT平均响应时间。
		max_select_elapsed	Long	SELECT最大响应时间。
		min_select_elapsed	Long	SELECT最小响应时间。
		total_update_elapsed	Long	UPDATE总响应时间。
		avg_update_elapsed	Long	UPDATE最大响应时间。
		max_update_elapsed	Long	UPDATE最小响应时间。
		min_update_elapsed	Long	UPDATE最小响应时间。
		total_delete_elapsed	Long	DELETE总响应时间。
		avg_delete_elapsed	Long	DELETE平均响应时间。
		max_delete_elapsed	Long	DELETE最大响应时间。
		min_delete_elapsed	Long	DELETE最小响应时间。
		total_insert_elapsed	Long	INSERT总响应时间。
		avg_insert_elapsed	Long	INSERT平均响应时间。
		max_insert_elapsed	Long	INSERT最大响应时间。
		min_insert_elapsed	Long	INSERT最小响应时间。
		delta_select_count	Long	单位时间SELECT数量差值。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		delta_update_count	Long	单位时间UPDATE数量差值。
		delta_insert_count	Long	单位时间INSERT数量差值。
		delta_delete_count	Long	单位时间DELETE数量差值。
		delta_mergeinto_count	Long	单位时间MERGEINTO数量差值。
		delta_ddl_count	Long	单位时间DDL数量差值。
		delta_dml_count	Long	单位时间DML数量差值。
		delta_dcl_count	Long	单位时间DCL数量差值。
		delta_total_select_elapse	Long	单位时间SELECT总响应时间差值。
		delta_avg_select_elapse	Long	单位时间SELECT平均响应时间差值。
		delta_max_select_elapse	Long	单位时间SELECT最大响应时间差值。
		delta_min_select_elapse	Long	单位时间SELECT最小响应时间差值。
		delta_total_update_elapse	Long	单位时间UPDATE总响应时间差值。
		delta_avg_update_elapse	Long	单位时间UPDATE最大响应时间差值。
		delta_max_update_elapse	Long	单位时间UPDATE最小响应时间差值。
		delta_min_update_elapse	Long	单位时间UPDATE最小响应时间差值。
		delta_total_delete_elapse	Long	单位时间DELETE总响应时间差值。
		delta_avg_delete_elapse	Long	单位时间DELETE平均响应时间差值。
		delta_max_delete_elapse	Long	单位时间DELETE最大响应时间差值。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		delta_min_delete_elapse	Long	单位时间DELETE最小响应时间差值。
		delta_total_insert_elapse	Long	单位时间INSERT总响应时间差值。
		delta_avg_insert_elapse	Long	单位时间INSERT平均响应时间差值。
		delta_max_insert_elapse	Long	单位时间INSERT最大响应时间差值。
		delta_min_insert_elapse	Long	单位时间INSERT最小响应时间差值。
OsStat	系统状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		tcp_resend_rate	Float	tcp协议栈重传率%。
SchemaUsage	schema使用统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		db_name	String	数据库名称。
		schema_name	String	schema名称。
		used_space	Long	已使用容量（单位：Byte）。
		perm_space	Long	总容量（单位：Byte）。
SessionStats	session统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		db_name	String	数据库名称。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		inst_name	String	实例名称，为了说明当前值从哪个CN上取到的。
		pid	String	线程ID（可以作为sessionID，也可以作为连接ID使用）。
		user_name	String	数据库用户名。
		application_name	String	用户应用名称。
		client_address	String	客户端地址。
		client_hostname	String	客户端主机名。
		client_port	String	客户端用于与后台通讯的tcp端口号，如果使用unix套接字，则为-1。
		backend_start	Long	该过程开始的时间，即当客户端连接服务器时。
		elapsed_time	Long	到目前为止的执行时间。
		curr_xact_start	Long	启动当前事务的时间，如果没有事务是活跃的，则为null。如果当前查询是首个事务，则这列等同于query_start列。
		state_change	Long	上次状态改变的时间。
		waiting	Boolean	如果后台当前正等待锁则为true。
		state	String	该后台当前总体状态。
		resource_pool	String	用户使用的资源池。
		query_elapsed_time	Long	语句当前为止的实际执行时间（单位：s）。
		query_id	String	查询语句的ID。
		query_start	Long	语句执行的开始时间。
		control_group	String	语句当前所使用的cgroups。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		lock_type	String	锁类型。
		lock_mode	String	锁模式。
		lock_granted	Boolean	存在锁等待情况，是否持有锁，持有为true。
		lock_relname	String	锁等待的资源。
		stmt_type	String	语句类型。
		query	String	查询SQL。
		system_query	Boolean	是否为系统查询。
		query_band	String	查询计划。
TableDirtyPageRate	表脏页率统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		db_name	String	数据库名称。
		schema_name	String	模式名称。
		table_name	String	表名称（全名）。
		table_owner	String	表所有者。
		table_size	Long	表大小（字节）。
		dirty_page_rate	Double	表脏页率。
TableSkewRate	表倾斜监控统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		db_name	String	数据库名称。
		schema_name	String	模式名称。
		table_name	String	表名称（全名）。
		table_owner	String	表所有者。
		table_size	Long	表大小（字节）。
		skew_rate	Double	表倾斜率。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
WorkloadQueue	资源池统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		respool_name	String	负载资源池。
		cpu_quota	Integer	资源池CPU配额。
		mem_quota	Integer	资源池内存配额。
		disk_quota	Long	资源池磁盘配额。
		max_simple_query	Integer	资源池允许的简单作业最大并发数。
		max_complicated_query	Integer	资源池允许的最大并发数。
		cpu_usage	Double	资源池CPU使用率。
		mem_usage	Double	资源池内存使用率。
		disk_usage	Double	资源池磁盘使用率。
		simple_query	Integer	资源池当前简单作业并发数。
complicated_query	Integer	资源池当前并发数。		
business_complicated_simple_sql	根据SQL属性统计SQL数量	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		attribute	String	SQL属性。
		count	Integer	sql数量。
business_concurrency	业务并发	ctime	Timestamp	采集时间。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		user_name	String	用户名称。
		coorname	String	cn名称。
		state	String	状态。
		count	Integer	数量。
business_DN_thread_wait	业务DN等待状态计数	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		node_name	String	节点名称。
		count	Integer	数量。
business_enqueue	业务排队状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		user_name	String	用户名称。
		respool	String	资源池名称。
		queue	String	队列名称。
		count	Integer	数量。
business_hudi_sync_state	hudi同步任务提交时间	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		task_id	String	任务ID。
		latest_sync_end	Timestamp	最后同步数据。
		latest_commit	String	最后提交信息。
		latest_sync_count	Long	最后同步数量。
business_hudi_sync_state_log	hudi同步任务提交日志	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		task_id	String	任务ID。
		last_log	String	最后任务日志。
business_obs_latency	业务obs桶访问时延	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		node_name	String	节点名称。
		latency_ms	Float	logtime之前10分钟内OBS的平均延迟（单位：ms）。
		req_count	Long	logtime之前10分钟内OBS的请求次数。
		log_time	Timestamp	记录延迟信息的时刻。
business_obs_read_traffic	业务obs桶读流量	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		node_name	String	节点名称。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		traffic_mb	Float	logtime之前10分钟OBS读流量统计。
		bandwidth_mb_per_s	Float	平均带宽（单位：MB/s）。
		req_count	Long	logtime之前10分钟OBS读次数。
		log_time	Timestamp	记录统计信息时刻。
business_obs_write_traffic	业务obs桶写流量	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		node_name	String	节点名称。
		traffic_mb	Float	logtime之前10分钟OBS写流量统计。
		bandwidth_mb_per_s	Float	平均带宽（单位：MB/s）。
		req_count	Long	logtime之前10分钟OBS写次数。
		log_time	Timestamp	记录统计信息时刻。
business_query_monitor	业务实时语句	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		db_name	String	数据库名称。
		coorname	String	CN名称。
		pid	Long	线程ID。
		query_id	Long	查询ID。
		query_band	String	查询作业类型。
		user_name	String	用户名称。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		application_name	String	应用名称。
		client_address	String	客户端IP地址。
		client_hostname	String	客户端主机名。
		client_port	Integer	客户端用于与后端通讯的TCP端口号。
		waiting	String	如果后端当前正等待锁则为true。
		q_running	Long	执行时长。
		wlm_start	Long	语句执行的开始时间。
		block_time	Long	语句执行前的阻塞时间（单位：ms）。
		duration	Long	语句已经执行的时间（单位：ms）。
		enqueue	String	工作负载管理资源状态。
		resource_pool	String	用户使用的资源池。
		control_group	String	语句所使用的Cgroup。
		max_peak_memory	Integer	语句在所有DN上的最大内存峰值（单位：MB）。
		average_peak_memory	Integer	语句执行过程中的内存使用平均值（单位：MB）。
		memory_skew_percent	Integer	语句在各DN间的内存使用倾斜率。
		estimate_memory	Integer	语句预估使用内存（单位：MB）。
		spill_info	String	语句在所有DN上的下盘信息。
		max_spill_size	Integer	若发生下盘，所有下盘DN的最大下盘数据量（MB），默认为0。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		average_spill_size	Integer	若发生下盘，所有下盘DN的平均下盘数据量（MB），默认为0。
		spill_skew_percent	Integer	若发生下盘，DN间下盘倾斜率。
		max_dn_time	Long	语句在所有DN上的最大执行时间（单位：ms）。
		average_dn_time	Long	语句在所有DN上的平均执行时间（单位：ms）。
		dntime_skew_percent	Integer	语句在各DN间的执行时间倾斜率。
		max_cpu_time	Long	语句在所有DN上的最大CPU时间（单位：ms）。
		total_cpu_time	Long	语句在所有DN上的CPU总时间（单位：ms）。
		cpu_skew_percent	Integer	语句在各DN间的CPU时间倾斜率。
		warning	String	SQL自诊断调优相关告警。
		max_peak_iops	Integer	语句在所有DN上的每秒最大I/O峰值（列存单位是次/s，行存单位是万次/s）。
		average_peak_iops	Integer	语句在所有DN上的每秒平均I/O峰值（列存单位是次/s，行存单位是万次/s）。
		iops_skew_percent	Integer	语句在DN间的I/O倾斜率。
		query	String	正在执行的语句。
		node_group	String	语句所属用户对应的逻辑集群。
		query_status	String	后端当前总体状态。
		wlm_status	String	语句当前的状态。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		wlm_attrib	String	语句的属性。
		lane	String	表示语句查询的快慢车道。
		priority	Long	语句所在Cgroups的优先级。
		stream_cnt	Long	streaming线程的层级大于0的DN数量。
		dn_cnt	Long	streaming线程的层级等于0的DN数量。
		cn_cnt	Long	CN数量。
business_table_analyze	业务表统计信息状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		dbname	String	数据库名称。
		schema_name	String	schema名称。
		table_name	String	表名称。
		total_size	Long	表大小。
		last_analyze_time	Timestamp	表最后analyze时间。
business_table_skew_rate	表倾斜率	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		db_name	String	数据库名称。
		schema_name	String	schema名称。
		table_name	String	表名称。
		table_size	Long	表大小。
		skew_rate	Float	表倾斜率。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
business_thread_wait	业务等待事件	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		wait_status	String	等待状态。
		wait_event	String	等待事件。
		count	Integer	数量。
business_active_ratio_warn	业务并发使用率	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		nodename	String	节点名称。
		active	Integer	活跃数量。
		used_ratio	Float	使用占比。
business_conflict_lock	冲突的锁信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		locktype	String	被锁定对象的类型。
		nodename	String	被锁定对象的节点的名称。
		dbname	String	被锁定对象的数据库的名称。
		nspname	String	被锁定对象的命名空间的名称。
		relname	String	被锁定对象对应的关系的名称。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		partname	String	被锁定对象对应的分区的名称。
		page	Long	被锁定对象对应的页面的编号。
		tuple	Long	被锁定对象对应的元组的编号。
		transactionid	String	被锁定对象对应的事务的ID。
		username	String	申请锁的用户的名称。
		gxid	String	申请锁的事务的ID。
		xactstart	Timestamp	申请锁的事务的开始时间。
		queryid	Long	申请锁的线程的最新查询ID。
		query	String	申请锁的线程的最新查询语句。
		pid	Long	申请锁的线程的ID。
		mode	String	锁的级别。
		granted	Boolean	是否持有锁。
bussiness_connection_ratio	业务会话使用率	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		coorname	String	CN节点名称。
		active	Integer	活跃链接数量。
		setting	Integer	最大链接数量。
		used_ratio	Float	使用链接占比。
bussiness_resource_pool_info	资源池作业运行信息及资源使用信息	ctime	Timestamp	采集时间。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		rpname	String	该查询关联的资源池名称。
		nodegroup	String	执行该查询的集群名称。
		cn_count	Long	集群包含的CN数量。
		short_acc	Boolean	资源池是否开启短查询加速。
		session_count	Long	关联该资源池的会话数量。
		active_count	Long	关联该资源池的活跃会话数量。
		global_wait	Long	关联该资源池的所有作业中，因单CN上并发超max_active_statements引起排队的作业数。
		fast_run	Long	关联该资源池的所有作业中，正在资源池快车道运行的作业数。
		fast_wait	Long	关联该资源池的所有作业中，在资源池快车道排队的作业数。
		fast_limit	Long	资源池快车道作业并发上限。
		slow_run	Long	关联该资源池的所有作业中，正在资源池慢车道运行的作业数。
		slow_wait	Long	关联该资源池的所有作业中，在资源池慢车道排队的作业数。
		slow_limit	Long	资源池慢车道作业并发上限。
		used_mem	Float	资源池在所有DN上已用内存的平均值。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		estimate_mem	Long	资源池正在运行的作业估算内存之和。
		mem_limit	Float	资源池可用内存的上限。
		query_mem_limit	String	资源池内单个查询可以使用的内存上限。
		used_cpu	Float	资源池在所有DN上占用CPU核数的平均值。
		cpu_limit	Long	资源池在所有节点上可用CPU上限的平均值。
		read_speed	Float	资源池在所有DN上逻辑I/O读速率的平均值。
		write_speed	Float	资源池在所有DN上逻辑I/O写速率的平均值。
bussiness_resource_pool_user_info	用户作业运行信息及资源使用信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		username	String	用户名称。
		rpname	String	用户关联的资源池名称。
		nodegroup	String	资源池所属逻辑集群的名称。
		session_count	Long	资源该用户发起的会话数量。
		active_count	Long	该用户发起的会话数量。
		global_wait	Long	该用户执行的所有作业中，因单CN上并发超max_active_statements引起排队的作业数。
		fast_run	Long	该用户执行的所有作业中，正在资源池快车道运行的作业数。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		fast_wait	Long	该用户执行的所有作业中，在资源池快车道排队的作业数。
		slow_run	Long	该用户执行的所有作业中，正在资源池慢车道运行的作业数。
		slow_wait	Long	该用户执行的所有作业中，在资源池慢车道排队的作业数。
		used_mem	Float	用户在所有DN上已用内存的平均值，单位：MB。
		estimate_memory	Long	用户正在运行的作业估算内存之和，单位：MB。
		used_cpu	Float	用户在所有DN上使用CPU核数的平均值。
		read_speed	Long	用户在所有DN上逻辑IO读速率的平均值，单位：KB/s。
		write_speed	Long	用户在所有DN上逻辑IO写速率的平均值，单位：KB/s。
		used_space	Long	用户永久表已使用的空间大小，单位：KB。
		space_limit	Long	用户永久表可使用的空间大小上限，单位：KB。
		used_temp_space	Long	用户临时表已使用的空间大小，单位：KB。
		temp_space_limit	Long	用户临时表可使用的空间大小上限，单位：KB。
		used_spill_space	Long	用户中间结果集落盘已使用的空间大小，单位：KB。
		spill_space_limit	Long	用户中间结果集落盘可使用的空间大小上限，单位：KB。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
cluster_snap_shot_status	当前GTM上事务信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		xact_min	String	仍在运行的最小事务号。
		xact_max	String	已完成的事务号最大的事务的下一个事务号。
		xact_csn	String	待提交事务的序列号。
		xact_oldestxmin	String	当前最早的活跃事务在其取快照时，所有运行事务号最小的事务。
		xact_cnt	String	当前活跃的事务个数。
		xact_running_xids	String	当前活跃的事务号。
cluster_top_sql_size	TopSql大小	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		size	Long	TopSql大小。
database_cu_check	列存小CU	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		database	String	数据库名称。
		schema	String	schema名称。
		tablename	String	表名称。
		tablesize	Long	表大小。
		rows_per_cu	Long	列存小CU。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
database_db_monitor_probe	语句执行情况	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		create_cost	Float	建表耗时。
		insert_cost	Float	插入耗时。
		select_cost	Float	select耗时。
		drop_cost	Float	删表耗时。
		commit_cost	Float	事务提交耗时。
		create_detail	String	建表的执行计划。
		insert_detail	String	插入的执行计划。
		select_detail	String	select的执行计划。
		drop_detail	String	删表的执行计划。
database_replication_slots	复制节点的信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		slot_name	String	复制节点的名称。
		slot_type	String	复制节点的类型。
		active	Boolean	复制节点是否为激活状态。
		restart_lsn	String	复制节点的Xlog文件信息。
database_system_tables_size	数据库表信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		table_name	String	表名称。
		table_size	Integer	表大小。
database_user_partitions_count	数据库用户表分区总数	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		database	String	数据库名。
		partitions_count	Integer	分区数。
database_user_tables_count	用户表统计总数	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		database	String	数据库名。
		column_store_table_count	Integer	列存表数量。
		row_store_table_count	Integer	行存表数量。
		hstore_table_count	Integer	hstore表数量。
		ts_table_count	Integer	ts表数量。
		foreign_table_count	Integer	外表数量。
		view_count	Integer	hash表数量。
		index_count	Integer	replication表数量。
		sequence_count	Integer	polling表数量。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		hash_table_count	Integer	schema数量。
		replication_table_count	Integer	view数量。
		polling_table_count	Integer	proc数量。
		schema_count	Integer	index数量。
		proc_count	Integer	sequence数量。
database_wal_sender	DN上所有的WAL复制发送线程的统计信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		nodename	String	节点名称。
		sender_pid	Long	当前sender的pid。
		local_role	String	本地的角色。
		peer_role	String	对端的角色。
		peer_state	String	对端的状态。
		state	String	当前sender的复制状态。
		sender_sent_location	String	发送端发送的LSN位置。
		sender_write_location	String	发送端write的LSN位置。
		sender_flush_location	String	发送端flush的LSN位置。
		sender_replay_location	String	发送端replay的LSN位置。
		receiver_received_location	String	接收端received的LSN位置。
		receiver_write_location	String	接收端write的LSN位置。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		receiver_flush_location	String	接收端flush的LSN位置。
		receiver_replay_location	String	接收端replay的LSN位置。
inst_tmp_count	实例临时文件下盘信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		query_id	Long	造成下盘的SQL语句的query_ID。
		query	String	造成下盘的SQL语句。
		files_count	Long	下盘文件数量统计。
instance_checkpoint	实例checkpoint信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		label	String	信息属性。
		value	String	信息值。
instance_memory_warn	实例内存使用率	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		nodename	String	节点名称。
		process_used_memory	Long	进程使用内存。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		max_process_memory	Long	进程最大内存。
		used_ratio	Long	内存使用率。
instance_process_stat	实例进程 proc/stat 信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		insttype	String	实例类型。
		pid	Long	进程pid。
		command	String	task_struct结构体的进程名。
		state	String	进程状态，此处为S。
		ppid	Long	父进程ID（父进程是指通过fork方式，通过clone并非父进程）。
		pgrp	Long	进程组ID。
		session	Long	进程会话组ID。
		tty_nr	Long	当前进程的tty终点设备号。
		tpgid	Long	控制进程终端的前台进程号。
		flags	Long	程标识位，定义在“include/linux/sched.h”中的“PF_*”。
		minflt	Long	次要缺页中断的次数，即无需从磁盘加载内存页。比如COW和匿名页。
cminflt	Long	当前进程等待子进程的minflt。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		majflt	Long	主要缺页中断的次数，需要从磁盘加载内存页，比如map文件。
		majflt_cow	Long	当前进程等待子进程的majflt。
		utime	Long	该进程处于用户态的时间。
		stime	Long	该进程处于内核态的时间。
		cutime	Long	当前进程等待子进程的utime。
		cstime	Long	当前进程等待子进程的utime。
		priority	Long	进程优先级。
		nice	Long	nice值，取值范围[-20,19]。
		num_threads	Long	线程个数。
		itrealvalue	Long	该字段已废弃，恒等于0。
		starttime	Long	自系统启动后的进程创建时间。
		vsize	Long	进程的虚拟内存大小。
		rss	Long	进程独占内存+共享库（单位：pages）。
		rsslim	String	rss大小上限。
instance_space	实例内部数据空间统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		pg_xlog	Long	单实例xlog文件大小。
		pg_cbm	Long	单实例cbm文件大小。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		pg_rewind_bak	Long	单实例pg_rewind_bak文件大小。
		pgsql_tmp	Long	单实例下盘文件数量。
		core	Long	单实例core文件大小。
		coreowner	String	core文件属主。
node_agent_stat	节点dmsagent进程信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		insttype	String	实例类型CN/DN/GTM/CMA。
		pid	Long	进程pID。
		command	String	task_struct结构体的进程名。
		state	String	进程状态，此处为S。
		ppid	Long	父进程ID（父进程是指通过fork方式，通过clone并非父进程）。
		pgrp	Long	进程组ID。
		session	Long	进程会话组ID。
		tty_nr	Long	当前进程的tty终点设备号。
		tpgid	Long	控制进程终端的前台进程号。
		flags	Long	程标识位，定义在“include/linux/sched.h”中的“PF_*”。
minflt	Long	次要缺页中断的次数，即无需从磁盘加载内存页，比如COW和匿名页。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		cminflt	Long	当前进程等待子进程的minflt。
		majflt	Long	主要缺页中断的次数，需要从磁盘加载内存页，比如map文件。
		majflt_cow	Long	当前进程等待子进程的majflt。
		utime	Long	该进程处于用户态的时间。
		stime	Long	该进程处于内核态的时间。
		cutime	Long	当前进程等待子进程的utime。
		cstime	Long	当前进程等待子进程的utime。
		priority	Long	进程优先级。
		nice	Long	nice值，取值范围[-20,19]。
		num_threads	Long	线程个数。
		itrealvalue	Long	该字段已废弃，恒等于0。
		starttime	Long	自系统启动后的进程创建时间。
		vsize	Long	进程的虚拟内存大小。
		rss	Long	进程独占内存+共享库（单位：pages）。
		rsslim	String	rss大小上限。
node_fd_total_count	节点file descriptor文件描述符总量统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		alloc_count	Long	已分配文件句柄的数目。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		alloc_unuse_count	Long	已分配未使用文件句柄的数目。
		total_count	Long	文件句柄的最大数目。
node_fs_size	文件系统大小	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		filesystem	String	设备符。
		total	Long	总大小。
		used	Long	已使用大小。
		available	Long	可用大小。
		usage_percent	Integer	使用率。
		mounted_on	String	路径。
node_meminfo	节点/proc/meminfo信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		mem_type	String	/proc/meminfo中的第一列信息。
		value_in_kb	String	/proc/meminfo中的具体类型的mem数值。
node_memory_page	节点内存活动	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		pagein_rate	Float	磁盘每秒pagein的速率。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		pageout_rate	Float	磁盘每秒pageout的速率。
		pagefault_rate	Float	系统每秒pagefault速率。
		majfault_rate	Float	系统每秒majorpagefault速率。
		pagefree_rate	Float	系统每秒清理空闲页。
		pagescank_rate	Float	kswapd任务每秒扫描量。
		pgscand_rate	Float	系统直接每秒扫描量。
		pagesteal_rate	Float	系统pagecache回收速率。
		vmeff_ratio	Float	通过pgsteal/pgscan计算量。
node_memory_swap	节点内存swap状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		kbmemfree	Long	空闲内存（单位：KB）。
		kbmemused	Long	已使用内存（单位：KB）。
		memused_ratio	Float	内存使用率。
		kbuffers	Long	buffer内存（单位：KB）。
		kbcached	Long	cached内存（单位：KB）。
		kbcommit	Long	workload内存KBRAM/SWAP。
		commit_ratio	Float	workload内存占比。
		kbactive	Long	活跃内存（单位：KB）。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		kbinact	Long	非活跃内存（单位：KB）。
		kbdirty	Long	pagecache需要回写的脏页（单位：KB）。
node_net_stat	节点网络信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		interface	String	网络接口。
		rxpps	Float	收包数/s。
		txpps	Float	发包数/s。
		rxkpbs	Float	收包字节数/s。
		txkpbs	Float	发包字节数/s。
		rxcmpps	Float	压缩报文接收数/s。
		txcmpps	Float	压缩报文发送数/s。
		rxmstps	Float	组播包接收数量/s。
		ifutil	Float	使用率。
		rxerrps	Float	错包接收数/s。
		txerrps	Float	错包发送数/s。
		collps	Float	坏包数/s。
		rxdropps	Float	接收侧丢包/s。
		txdropps	Float	发送侧丢包/s。
		txcarrps	Float	发送侧ce包/s。
		rxframpps	Float	接收侧坏帧/s。
		rxfifops	Float	接收侧FIFO队列溢出/s。
txfifops	Float	发送侧FIFO队列溢出/s。		
tc_delay	Integer	tc延迟配置。		
tc_loss	Float	tc丢包率配置。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		tc_drop	Long	tc丢包数。
node_processes_count	节点进程状态	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		d_thread_count	Integer	d线程数量。
		z_thread_count	Integer	z线程数量。
		total_process_count	Integer	系统总进程数。
node_processes_CPU_top20	节点TOP20CPU进程	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		username	String	用户名。
		processid	Long	pID。
		cpu_percent	Float	CPU使用率。
		mem_percent	Float	内存使用率。
		vir_mem_size	String	虚拟内存。
		res_mem_size	String	内存使用量。
		stat	String	状态。
		start	String	启动时间。
		duration	String	运行时间。
node_processes_mem_top20	节点TOP20内存进程	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		host_id	Integer	采集上报的节点ID。
		username	String	用户名。
		processid	Long	pID。
		cpu_percent	Float	CPU使用率。
		mem_percent	Float	内存使用率。
		vir_mem_size	String	虚拟内存。
		res_mem_size	String	内存使用量。
		stat	String	状态。
		start	String	启动时间。
		duration	String	运行时间。
		command	String	执行cmd。
node_task_loadavg	节点负载信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		runq_size	Integer	任务队列。
		total_task_count	Integer	系统总任务数。
		loadavg1	Float	1min负载。
		loadavg5	Float	5min负载。
		loadavg15	Float	15min负载。
blocked	Integer	I/O阻塞数量。		
node_tcp_snmp	节点/proc/net/snmp信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		rtoalgorithm	Long	无意义。
		rtomin	Long	无意义。
		rtomax	Long	无意义。
		maxconn	Long	最大连接数。
		activeopens	Long	服务器主动连接的TCP请求数。
		passiveopens	Long	服务器接收到的TCP请求数。
		attemptfails	Long	TCP连接失败个数 (AttemptFails) 。
		estabresets	Long	无意义。
		currestab	Long	TCP当前建立连接的个数。
		insegs	Long	收到分片总数。
		outsegs	Long	发送分片总数。
		retranssegs	Long	重传数。
		inerrs	Long	接收错包数。
		outrsts	Long	发送错包数。
incsumerrors	Long	无意义。		
node_tcp_state_count	节点tcp连接状态统计	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		established	Integer	当前建联。
		synsent	Integer	当前处于sync_sent状态。
		synrecv	Integer	当前处于sync_recv状态。
		finwait1	Integer	当前处于finwait1状态。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		finwait2	Integer	当前处于finwait2状态。
		timewait	Integer	当前处于timewait状态。
		closed	Integer	当前处于closed状态。
		closewait	Integer	当前处于closewait状态。
		lastack	Integer	当前处于lastack状态。
		listening	Integer	当前处于listening状态。
		closing	Integer	当前处于closing状态。
warn_conflict_lock_duration	冲突的锁信息	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		locktype	String	被锁定对象的类型。
		nodename	String	被锁定对象的节点的名称。
		dbname	String	被锁定对象的数据库的名称。
		nspname	String	被锁定对象的命名空间的名称。
		relname	String	被锁定对象对应的关系的名称。
		partname	String	被锁定对象对应的分区的名称。
		page	Long	被锁定对象对应的页面的编号。
tuple	Long	被锁定对象对应的元组的编号。		

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		transactionid	String	被锁定对象对应的事务的ID。
		username	String	申请锁的用户的名称。
		gxid	String	申请锁的事务的ID。
		xactstart	Timestamp	申请锁的事务的开始时间。
		queryid	Long	申请锁的线程的最新查询ID。
		query	String	申请锁的线程的最新查询语句。
		pid	Long	申请锁的线程的ID。
		mode	String	锁的级别。
		duration	Long	是否持有锁。
warn_instance_checkpoint	实例checkpoint时间	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		instid	Integer	实例ID。
		last_cp_time	Long	最近的一次checkpoint时间。
		since_last_cp	Long	当前系统时间和last_cp_time的差值。
warn_node_memory	非DWS的任务系统内存开销	ctime	Timestamp	采集时间。
		virtual_cluster_id	Integer	集群虚拟ID。
		host_id	Integer	采集上报的节点ID。
		total_mem	Long	节点总内存。
		total_use	Long	节点已使用内存。
		instance_use	Long	实例已用内存。

指标名称	指标描述	指标返回参数	指标返回参数类型	指标返回参数描述
		os_ratio	Long	非gaussdb的系统使用内存占比。