

数据仓库服务

API 参考

文档版本 01

发布日期 2023-11-28



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

未经本公司书面许可，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
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束与限制.....	1
1.5 基本概念.....	1
2 API 概述.....	3
3 如何调用 API.....	4
3.1 构造请求.....	4
3.2 认证鉴权.....	7
3.3 返回结果.....	8
4 快速入门.....	10
5 API 说明.....	16
5.1 集群管理.....	16
5.1.1 创建集群.....	16
5.1.2 查询集群列表.....	23
5.1.3 查询集群详情.....	34
5.1.4 查询节点类型.....	47
5.1.5 删除集群.....	53
5.1.6 重启集群.....	57
5.1.7 扩容集群.....	60
5.1.8 重置密码.....	64
5.1.9 查询重分布详情.....	68
5.1.10 查询参数修改审计记录.....	75
5.1.11 保存集群描述信息.....	80
5.1.12 查询单个实例.....	84
5.2 快照管理.....	91
5.2.1 创建快照.....	91
5.2.2 查询快照列表.....	95
5.2.3 查询快照详情.....	100
5.2.4 删除手动快照.....	105
5.2.5 恢复集群.....	108

5.2.6 恢复表.....	114
5.2.7 获取跨区域快照可用 Region.....	119
5.2.8 设置跨区域备份配置.....	123
5.2.9 查询所有跨区域快照配置.....	127
5.2.10 删除跨区域备份配置.....	131
5.3 资源管理.....	134
5.3.1 启动资源管理计划.....	134
5.3.2 查询某个资源管理计划详细信息.....	138
5.3.3 停止资源管理计划.....	143
5.3.4 删除资源管理计划.....	147
5.4 主机监控.....	150
5.4.1 查询表倾斜或脏页率信息.....	150
5.4.2 查询 SQL 列表.....	155
5.4.3 查询 SQL 执行信息.....	178
5.5 日志管理.....	185
5.5.1 获取 LTS 日志列表.....	185
5.5.2 开启云服务日志.....	189
5.5.3 关闭云服务日志.....	193
5.6 升级管理.....	196
5.6.1 获取集群升级记录.....	196
5.6.2 获取集群可升级的目标版本.....	200
5.6.3 下发集群升级相关操作.....	204
6 应用示例.....	209
6.1 使用 Postman 调用创建集群接口.....	209
6.2 使用 Postman 调用创建快照接口.....	211
7 权限及授权项说明.....	213
8 附录.....	216
8.1 状态码.....	216
8.2 错误码.....	218
8.3 创建 VPC.....	226
8.4 获取项目 ID.....	226
8.5 获取帐号 ID.....	228
8.6 获取集群 ID.....	228

1 使用前必读

1.1 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。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](#)。

1.2 调用说明

数据仓库服务提供了REST (Representational State Transfer) 风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

1.3 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。

1.4 约束与限制

- 您能创建的数据仓库服务资源的数量与配额有关系，如果您想查看服务配额、扩大配额，具体请参见“[配额说明](#)”。
- 更详细的限制请参见具体API的说明。

1.5 基本概念

- 帐号

用户注册时的帐号，帐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。由于帐号是付费主体，为了确保帐号安全，建

议您不要直接使用帐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

由帐号在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）。

通常在调用API的鉴权过程中，您需要用到帐号、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。

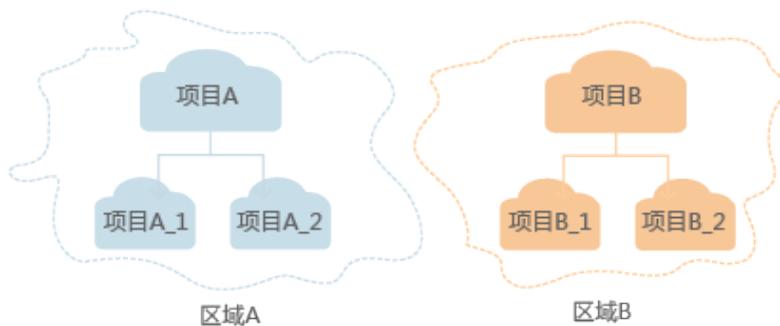
- 可用区

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，用户可以访问您帐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



- 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。

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。

类型	API	说明
集群管理 接口	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。
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。
	恢复集群	使用快照恢复集群。

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](#)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。

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。

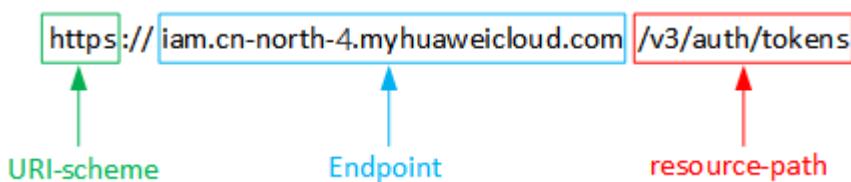
表 3-1 URI 中的参数说明

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IAM服务在“中国-香港”区域的Endpoint为“iam.ap-southeast-1.myhuaweicloud.com”。
resource-path	资源路径，也即API访问路径。从具体API的URI模块获取，例如“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“中国-香港”区域的Token，则需使用“中国-香港”区域的Endpoint（iam.ap-southeast-1.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.ap-southeast-1.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.ap-southeast-1.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。	注：以下仅为Token示例片段 MIIPAgYJKoZIhvcNAQcCo ...ggg1BBIINPXsidG9rZ
X-Language	请求语言。	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.ap-southeast-1.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*为用户所属的帐号名称，*****为用户登录密码，xxxxxxxxxxxxxxxxxxxx为project的名称，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的作用域为某个帐号下所有资源或帐号的某个project下的资源，详细定义请参见[获取用户Token](#)。

```
POST https://iam.ap-southeast-1.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。

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 → noopener
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": "ap-southeast-1", ..... } ] } ] }
```

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": "ap-southeast-1"  
            }  
        }  
    }  
' -v -k
```

1. 获取响应Header中“X-Subject-Token”的值即为Token，示例如下。

X-Subject-Token: MlidkgYJKoZlhvcNAQcColidgzCCA38CAQExDTALBglghkgBZQMEAeWgXXXX...

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=MlidkgYJKoZlhvcNAQcColidgzCCA38CAQExDTALBglghkgBZQMEAeWgXXXX...

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": "8"  
                }  
            ]  
        }  
    ]  
}
```

```
        "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": "ap-southeast-1",
            "create_time": "2023-11-28T10:00:00Z",
            "update_time": "2023-11-28T10:00:00Z"
        }
    ]
}
```

```
"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.ap-southeast-1.myhuaweicloud.com",
        "jdbc_url": "jdbc:postgresql://dws-demo.ap-southeast-1.myhuaweicloud.com/
<YOUR_DATABASE_name>"
    }
],
"endpoints": [
    {
        "connect_info": "dws-demo.ap-southeast-1.myhuaweicloud.com",
        "jdbc_url": "jdbc:postgresql://dws-demo.ap-southeast-1.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": "ap-southeast-1",
        "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.ap-southeast-1.myhuaweicloud.com",
                "jdbc_url": "jdbc:postgresql://dws-demo.ap-southeast-1.myhuaweicloud.com/
<YOUR_DATABASE_name>"
            }
        ]
    }
}
```

```
        "jdbc_url": "jdbc:postgresql://dws-demo.ap-southeast-1.myhuaweicloud.com/
<YOUR_DATABASE_name>"
    }
],
"endpoints": [
{
    "connect_info": "dws-demo.ap-southeast-1.myhuaweicloud.com",
    "jdbc_url": "jdbc:postgresql://dws-demo.ap-southeast-1.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以创建一个名称为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 集群管理

5.1.1 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。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/clusters

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

表 5-2 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 5-3 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 。
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。
user_name	是	String	DWS集群管理员用户名。用户名要求如下： <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。
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。
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 5-4 PublicIp

参数	是否必选	参数类型	描述
public_bind_type	是	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 5-5 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。

参数	是否必选	参数类型	描述
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 5-6 响应 Body 参数

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

表 5-7 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，带公网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-dbef7d9a720",
    "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("Passw0rd!")  
            .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="PasswOrd!",
            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"  
    "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{}  
    eiIpPublicIp:= ""  
    publicIpCluster := &model.PublicIp{  
        PublicBindType: "auto_assign",  
        EipId: &eiIpPublicIp,  
    }  
    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。

5.1.2 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 5-8 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 5-9 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 5-10 ClusterInfo

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
status	String	集群状态，有效值包括： <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。
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。

参数	参数类型	描述
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>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：</p> <pre>"action_progress": {"SNAPSHOTTING":"16%"} }</pre>

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_STANDBY_SWITCHOVER：主备恢复中

参数	参数类型	描述
		<ul style="list-style-type: none">• ACTIVE_STANDBY_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 5-11 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 5-12 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 5-13 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 5-14 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 5-15 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 5-16 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-4fc4-4e5a-a9dc-bb903abb494b",  
      "status": "200"  
    }, {  
      "id": "d71a4a25-c9bc-4ffd-9f4a-e422aef327f9",  
      "status": "200"  
    } ],  
    "tags": [ {  
      "key": "key1",  
      "value": "value1"  
    }, {  
      "key": "key2",  
      "value": "value2"  
    } ],  
    "user_name": "dbadmin",  
    "password": "*****"  
  } ]  
}
```

```
"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。

5.1.3 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 5-17 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 5-18 响应 Body 参数

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

表 5-19 ClusterDetail

参数	参数类型	描述
id	String	集群ID。
name	String	集群名称。
status	String	集群状态, 有效值包括: <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，详情请参加 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_STANDBY_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

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_STANDBY_SWITCHOVER：主备恢复中

参数	参数类型	描述
		<ul style="list-style-type: none">ACTIVE_STANDBY_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 5-20 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 5-21 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 5-22 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 5-23 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 5-24 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 5-25 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 5-26 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 5-27 ResizeInfo

参数	参数类型	描述
target_node_num	Integer	扩容后的节点数量。
origin_node_num	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 5-28 FailedReason

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

表 5-29 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-4fc4-4e5a-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-9etypc2a-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。

5.1.4 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 5-30 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 5-31 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 5-32 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。● stream: IoT数仓。
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 5-33 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 5-34 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 5-35 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 5-36 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 5-37 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 5-38 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",  
        "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。

5.1.5 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 5-39 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 5-40 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。

5.1.6 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 5-41 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 5-42 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.getHttpStatus());
            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。

5.1.7 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 5-43 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 5-44 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 5-45 ScaleOut

参数	是否必选	参数类型	描述
count	是	Integer	扩容节点数。

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.getHttpStatus());
        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
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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"
    "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。

5.1.8 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 5-46 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 。
project_id	是	String	项目ID。获取方法，请参见 获取项目ID 。

请求参数

表 5-47 请求 Body 参数

参数	是否必选	参数类型	描述
new_password	是	String	GaussDB(DWS) 集群管理员新密码。新密码复杂度要求如下： <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="NewPassw0rd!")
)
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: "NewPassw0rd!",
    }
    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。

5.1.9 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 5-48 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 5-49 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。

参数	是否必选	参数类型	描述
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 5-50 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 5-51 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。

参数	参数类型	描述
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 5-52 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 5-53 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 5-54 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。

参数	参数类型	描述
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。
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 5-55 RedisTableDetail

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

表 5-56 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。

参数	参数类型	描述
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>");  
        requestWithType("<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.getHttpStatus());  
            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。

5.1.10 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 5-57 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 5-58 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 5-59 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 5-60 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 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 = 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。

5.1.11 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 5-61 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 5-62 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 5-63 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 5-64 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。

5.1.12 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 5-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法, 请参见 获取项目ID 。
instance_id	是	String	实例ID。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 5-66 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 5-67 ClusterFlavorResp

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

表 5-68 CompatibleInstanceVolumeResp

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

表 5-69 CompatibleDataStoreResp

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

表 5-70 CompatibleFaultResp

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

表 5-71 CompatibleConfigurationResp

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

表 5-72 CompatibleReplicasResp

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

表 5-73 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.getHttpStatus());
            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 快照管理

5.2.1 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 5-74 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 5-75 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 5-76 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 5-77 响应 Body 参数

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

表 5-78 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.withName("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。
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。

5.2.2 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 5-79 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 5-80 响应 Body 参数

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

表 5-81 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。

参数	参数类型	描述
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-0e3bbcd8ab1",  
        "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
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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。

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。

5.2.3 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 5-82 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 5-83 响应 Body 参数

参数	参数类型	描述
snapshot	SnapshotDetail object	快照详情对象。

表 5-84 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。
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。

参数	参数类型	描述
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 5-85 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 5-86 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。

5.2.4 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 5-87 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。

5.2.5 恢复集群

功能介绍

该接口用于使用快照恢复集群。

调用方法

请参见[如何调用API](#)。

URI

POST /v1.0/{project_id}/snapshots/{snapshot_id}/actions

表 5-88 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法, 请参见 获取项目ID 。
snapshot_id	是	String	待恢复的快照ID。

请求参数

表 5-89 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 5-90 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。
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。

参数	是否必选	参数类型	描述
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。

表 5-91 PublicIp

参数	是否必选	参数类型	描述
public_bind_type	是	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 5-92 响应 Body 参数

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

表 5-93 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"
    "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",
        EipId: &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。

5.2.6 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 5-94 路径参数

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

请求参数

表 5-95 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 5-96 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 5-97 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。

5.2.7 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 5-98 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 5-99 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 5-100 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 5-101 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。

5.2.8 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 5-102 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 5-103 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。

5.2.9 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 5-104 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 5-105 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 5-106 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 5-107 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。
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.getHttpStatus());
            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
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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。

5.2.10 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 5-108 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 5-109 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 资源管理

5.3.1 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 5-110 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 5-111 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": ""  
}
```

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。

5.3.2 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 5-112 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 5-113 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 5-114 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 5-115 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。

参数	参数类型	描述
next_valid_time	String	下次校验时间。
queue_list	Array of QueueResourceItem objects	资源队列列表。

表 5-116 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 5-117 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",  
    "name": "test-plan-1",  
    "description": "Test Plan 1",  
    "start_time": "2023-11-28T10:00:00Z",  
    "end_time": "2023-11-28T12:00:00Z",  
    "interval": "10m",  
    "resource_items": [  
      {"resource_name": "cpu", "resource_value": 10, "value_unit": "cores"},  
      {"resource_name": "memory", "resource_value": 15, "value_unit": "GB"},  
      {"resource_name": "storage", "resource_value": 5, "value_unit": "TB"}  
    ]  
  }  
}
```

```
"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。

5.3.3 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 5-118 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 5-119 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": ""  
}
```

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.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 = 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。

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。

5.3.4 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 5-120 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 5-121 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 主机监控

5.4.1 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 5-122 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 5-123 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。
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 5-124 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 5-125 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_size": "100 MB",  
            "table_owner": "pmk",  
            "dirty_page_rate": 0.01,  
            "skew_rate": 0.05  
        }  
    ]  
}
```

```
"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.getHttpStatus());
            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。

5.4.2 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 5-126 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 5-127 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 5-128 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。

参数	是否必选	参数类型	描述
operator	是	String	<p>比较方式：</p> <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 5-129 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 5-130 ListQueriesData

参数	参数类型	描述
queries	Array of ListQueriesDto objects	查询数据列表。

参数	参数类型	描述
status	ListQueriesStatus object	统计数据。

表 5-131 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)。
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。

参数	参数类型	描述
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。
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）。
dntime_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）。

参数	参数类型	描述
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_attrib	String	语句的属性 (ordinary, simple, complicated, internal)
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 5-132 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",  
        "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": "="  
    }]
```

```
"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" : "<>"  
}, {  

```

```
    "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()
```

```
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()
```

```
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("averagePeakIops")
        .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.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 = ListQueriesRequest()
    listConditionsbody = [
        ListQueriesCondition(
            field="userName",
            value="用户名",
            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="",
            operator="="
        )
    ]
    response = client.list_queries(request)
    print(response)
except exceptions.SDKException as e:
    print(e)
```

```
        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="vlmStatus",  
    value="语句运行状态",  
    operator="="  
)  
,  
ListQueriesCondition(  
    field="vlmAttrib",  
    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",
        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: "averagePeakIops",
            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。

状态码	描述
503	服务不可用。

5.4.3 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 5-133 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 5-134 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 5-135 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	ListQueriesDt o object	响应数据。

表 5-136 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)。
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。

参数	参数类型	描述
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。
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）。

参数	参数类型	描述
dntime_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_attrib	String	语句的属性 (ordinary, simple, complicated, internal)
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)。

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": "145522562959855061",  
        "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.getHttpStatus());  
            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。

5.5 日志管理

5.5.1 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 5-137 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 5-138 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 5-139 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 5-140 LtslogInfo

参数	参数类型	描述
status	String	配置状态。
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。

5.5.2 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 5-141 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 。
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.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 = 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

5.5.3 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 5-142 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。获取方法, 请参见 获取项目ID 。
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
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment 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

状态码	描述
404	找不到资源
500	服务内部错误
503	服务不可用

5.6 升级管理

5.6.1 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 5-143 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 5-144 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 5-145 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 5-146 ClusterUpdateRecordResp

参数	参数类型	描述
item_id	String	升级项目ID。
status	String	升级状态。
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": "2023-08-04T06:46:46",  
            "job_id": "c719b1a7-c85c-4cb5-a721-7694908c2c11",  
            "failed_reason": null  
        }  
    ]  
}
```

```
        "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。

5.6.2 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 5-147 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 5-148 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 5-149 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 UpdateItemR esp objects	集群升级路径列表。

表 5-150 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。

5.6.3 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 5-151 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 5-152 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 5-153 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.withItemId("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.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 = 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

- 已注册华为云帐号并开通华为云。
- 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](#)。
- 已获取华为云帐号对应的用户名、用户密码、帐号名及项目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、帐号名、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。

- 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、帐号名及项目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。

- 单击右上角“Send”，发送“POST https://iam.ap-southeast-1.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[者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.ap-southeast-1.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、帐号名及项目ID。项目ID获取方法请参见[获取项目ID](#)。

说明

获取token的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.ap-southeast-1.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.ap-southeast-1.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)，如果华为云帐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。

帐号具备所有接口的调用权限，如果使用帐号下的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](#)。

说明书

“√”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	√	x
查询集群列表	GET /v1.0/{project_id}/clusters	dws:openAPICluster:list	√	x
查询集群详情	GET /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:getDetail	√	x
查询节点类型	GET /v2/{project_id}/node-types	dws:openAPIFlavors:get	√	x
删除集群	DELETE /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:delete	√	x
重启集群	POST /v1.0/{project_id}/clusters/{cluster_id}/restart	dws:openAPICluster:restart	√	x
扩容集群大小	POST /v1.0/{project_id}/clusters/{cluster_id}/resize	dws:cluster:scaleOutOrOpenAPIResize	√	x

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	√	✗

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	√	✗
恢复集群	POST /v1.0/{project_id}/snapshots/{snapshot_id}/actions	dws:openAPISnapshot: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	UnprocessableEntity	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.5 001	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.5 002	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.5 003	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.5 004	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.5 005	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.5 006	无效的规格。	无效的规格。	请输入正确的规格后重试。
400	DWS.5 010	无效区域。	无效区域。	请重新选择有效的区域后重试。
400	DWS.5 011	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.5 012	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.5 013	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.5 014	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.5 015	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.5 016	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.5 051	ip非法。	ip非法。	请输入正确的ip后重试。
400	DWS.5 052	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.5 053	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.5 054	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.5 056	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.5 057	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.5 059	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.5 060	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.5 061	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.5 062	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.5 063	Specific字段无效。	Specific字段无效。	请输入正确的Specific值后重试。
400	DWS.5 064	UUID非法。	UUID非法。	请输入正确的UUID后重试。
400	DWS.5 065	格式非法。	格式非法。	请输入正确的格式后重试。
400	DWS.5 070	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.5 071	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.5 078	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.5 079	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值后重试。
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.5 200	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.5 202	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.5 203	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.5 206	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": "ap-southeast-1",  
            "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“我的凭证”页面的项目列表中查看项目ID。

图 8-1 查看项目 ID



8.5 获取帐号 ID

在调用接口的时候，部分URL中需要填入帐号ID（domain-id），所以需要先在管理控制台上获取到帐号ID。帐号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“我的凭证”页面中查看帐号ID。

图 8-2 获取帐号 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-4fc4-4e5a-a9dc-bb903abb494b",  
          "status": "200"  
        }  
      ]  
    }  
  ]  
}
```

```
        "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

The screenshot shows the 'Cluster Details' page in the DWS console. On the left, there's a sidebar with various tabs like 'Cluster Overview', 'Resource Management', 'Smart Maintenance', etc. The main area has two sections: 'Basic Information' and 'Network'. In 'Basic Information', the 'Cluster ID' field is highlighted with a red box and contains the value '15d19a62-caf8-4e8c-9d95-3d2b91e964a8'. Other fields include 'Cluster Name' (14), 'Cluster Status' (可用 - Available), 'Parameter Configuration Status' (已同步 - Synced), 'Maintenance Window' (周五 06:00-10:00 GMT+08:00), 'Enterprise Project' (default), 'Spec Details' (Standard Edition | 4 vCPUs | 32 GB Memory | 100 GB High I/O), and 'Cluster Description' (empty). In the 'Network' section, it shows 'Region' (Ulanqab · 二零三), 'Virtual Private Cloud' (vpc-6d93-0310), 'Subnet' (subnet-d83f), and 'Security Groups' (empty).

----结束