MapReduce 服务

API 参考

文档版本 01

发布日期 2024-12-27

华为云计算技术有限公司

商标声明


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

欢迎使用MapReduce服务（MapReduce Service，MRS）。MRS服务提供租户完全可控的企业级大数据集群云服务，轻松运行Hadoop、Spark、HBase、Kafka、Storm等大数据组件。

您可以使用本文档提供API对MRS服务进行相关操作，如创建集群、删除集群、调整集群节点、创建作业并执行等。支持的全部操作请参见**API概览**。

在调用MRS服务API之前，请确保已经充分了解MRS服务相关概念，详细信息请参见**产品介绍**。

MRS 3.x版本镜像，不支持MRS V1.1作业管理接口，需要使用V2作业管理接口。如果仍需使用V1.1作业管理接口，请在**历史API**中，获取相关接口。

终端节点

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

基本概念

- 账号

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

- 用户

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

  在**我的凭证**下，您可以查看项目ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

  从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

  详情请参见**区域和可用区**。

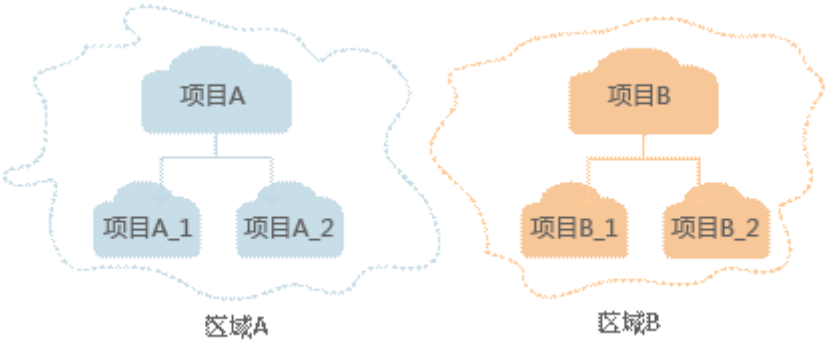
- 可用区（AZ，Availability Zone）

  一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 项目

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

  图 1-1 项目隔离模型

  同样在**我的凭证**下，您可以查看项目ID。

- 企业项目

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

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

2 API 概览

MRS提供的符合RESTful API的设计规范的接口，如表2-1和表2-2所示。如何选择不同版本的接口请参见API版本选择建议。

MRS版本演进过程中逐步废弃部分接口，具体请参见表2-3。

表 2-1 V2 接口

接口	功能	API URI
集群管理接口	创建集群	POST /v2/{project_id}/clusters
	修改集群名称	PUT /v2/{project_id}/clusters/{cluster_id}/cluster-name
	创建集群并提交作业	POST /v2/{project_id}/run-job-flow
	扩容集群	POST /v2/{project_id}/clusters/{cluster_id}/expand
	缩容集群	POST /v2/{project_id}/clusters/{cluster_id}/shrink
	集群添加组件	POST /v2/{project_id}/clusters/{cluster_id}/components
	查询集群节点列表	GET /v2/{project_id}/clusters/{cluster_id}/nodes
作业对象接口	新增并执行作业	POST /v2/{project_id}/clusters/{cluster_id}/job-executions
	查询作业列表信息	GET /v2/{project_id}/clusters/{cluster_id}/job-executions
	查询单个作业信息	GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}
	终止作业	POST /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/kill

接口	功能	API URI
	获取SQL结果	GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/sql-result
	批量删除作业	POST /v2/{project_id}/clusters/{cluster_id}/job-executions/batch-delete
弹性伸缩接口	查看弹性伸缩策略	GET /v2/{project_id}/autoscaling-policy/{cluster_id}
	更新弹性伸缩策略	PUT /v2/{project_id}/autoscaling-policy/{cluster_id}
	删除弹性伸缩策略	DELETE /v2/{project_id}/autoscaling-policy/{cluster_id}
	创建弹性伸缩策略	POST /v2/{project_id}/autoscaling-policy/{cluster_id}
集群HDFS文件接口	获取指定目录文件列表	GET /v2/{project_id}/clusters/{cluster_id}/files?path={directory}&offset={offset}&limit={limit}&sort_key={sort_key}&order={order}
SQL接口	提交SQL语句	POST /v2/{project_id}/clusters/{cluster_id}/sql-execution
	查询SQL结果	GET /v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}
	取消SQL执行任务	POST /v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}/cancel
委托管理（V2）	查询用户（组）与IAM委托的映射关系	GET /v2/{project_id}/clusters/{cluster_id}/agency-mapping
	更新用户（组）与IAM委托的映射关系	PUT /v2/{project_id}/clusters/{cluster_id}/agency-mapping
数据连接管理	创建数据连接	POST /v2/{project_id}/data-connectors
	查询数据连接列表	GET /v2/{project_id}/data-connectors
	更新数据连接	PUT /v2/{project_id}/data-connectors/{connector_id}
	删除数据连接	DELETE /v2/{project_id}/data-connectors/{connector_id}
版本元数据查询	展示MRS版本列表	GET /v2/{project_id}/metadata/versions

接口	功能	API URI
	查询MRS集群版本可用的规格	GET /v2/{project_id}/metadata/version/{version_name}/available-flavor
IAM同步管理接口	获取已经同步的IAM用户和用户组	GET /v2/{project_id}/clusters/{cluster_id}/iam-sync-user
	IAM同步	POST /v2/{project_id}/clusters/{cluster_id}/iam-sync-user
	指定用户、用户组取消同步	DELETE /v2/{project_id}/clusters/{cluster_id}/iam-sync-user
标签管理接口	集群操作默认标签	POST /v2/{project_id}/clusters/{cluster_id}/tags/switch
	查询默认标签状态	GET /v2/{project_id}/clusters/{cluster_id}/tags/status
	查询标签配额	GET /v2/{project_id}/clusters/{cluster_id}/tags/quota

表 2-2 V1.1 接口

接口	功能	API URI
集群管理接口	创建集群并执行作业	POST /v1.1/{project_id}/run-job-flow
	调整集群节点	PUT /v1.1/{project_id}/cluster_infos/{cluster_id}
	查询集群列表	GET /v1.1/{project_id}/cluster_infos
	查询集群详情	GET /v1.1/{project_id}/cluster_infos/{cluster_id}
	查询主机列表	GET /v1.1/{project_id}/clusters/{cluster_id}/hosts
	删除集群	DELETE /v1.1/{project_id}/clusters/{cluster_id}
弹性伸缩接口	配置弹性伸缩规则	POST /v1.1/{project_id}/autoscaling-policy/{cluster_id}
标签管理接口	给指定集群添加标签	POST /v1.1/{project_id}/clusters/{cluster_id}/tags
	查询指定集群的标签	GET /v1.1/{project_id}/clusters/{cluster_id}/tags
	删除指定集群的标签	DELETE /v1.1/{project_id}/clusters/{cluster_id}/tags/{key}

接口	功能	API URI
	批量添加集群标签	POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action
	批量删除集群标签	POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action
	查询所有标签	GET /v1.1/{project_id}/clusters/tags
	查询特定标签的集群列表	POST /v1.1/{project_id}/clusters/resource_instances/action
可用区	查询可用区信息	GET /v1.1/{region_id}/available-zones
版本元数据查询	查询对应版本元数据	GET /v1.1/{project_id}/metadata/versions/{version_name}

表 2-3 历史 API 接口

接口	功能	API URI
V1.1 作业管理接口	新增作业并执行	POST /v1.1/{project_id}/jobs/submit-job
	查询作业exe对象列表	GET /v1.1/{project_id}/job-exes
	查询作业exe对象详情	GET /v1.1/{project_id}/job-exes/{job_exe_id}
	删除作业执行对象	DELETE /v1.1/{project_id}/job-executions/{job_execution_id}

3 API 版本选择建议

当前MRS服务对外API提供云服务自定义规范的API V1.1和V2两类接口，V2版本目前仅部分接口支持，主要用于提交作业和提交SQL语句。在接口功能相同的情况下，推荐您优先使用V2接口。

- MRS所有版本均支持V1.1接口。
- 针对MRS 1.X版本的集群，MRS 1.8.7及之后版本支持V2接口。
- 针对MRS 2.X版本的集群，MRS 2.0.3及之后版本支持V2接口。

在某些功能上V2接口以V1.1接口为基础，在功能上做了如下功能增强：

- 支持安全集群提交作业。
- 支持HiveSql、Spark python和Flink作业。
- 支持SparkSql和SparkScript结果查询。

整体API及对应功能列表详见**API概览**。

4 如何调用 API

4.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的**管理员创建IAM用户**说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

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

请求 URI

请求URI由如下部分组成：

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

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

- **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**终端节点**中获取。例如IAM服务在“华北-北京四”区域的Endpoint为“iam.cn-north-4.myhuaweicloud.com”。
- **resource-path**：资源路径，也即API访问路径。从具体接口的URI模块获取，例如从具体API的URI模块获取，例如“管理员创建IAM用户”接口的resource-path为“/v3.0/OS-USER/users”。
- **query-string**：查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

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

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

图 4-1 URI 示意图

说明

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

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

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

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

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

请求消息头

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

如下公共消息头需要添加到请求中。

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

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

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

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

请求消息体

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

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

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

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

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

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

到这里为止这个请求需要的内容就齐全了，您可以使用curl、postman或直接编写代码等方式发送请求调用API。

4.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。
- Token认证：通过Token认证通用请求。

AK/SK 认证

说明

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

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

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

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

须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

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

Token 认证

说明

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

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

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

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

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

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

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

Content-Type: application/json
X-Auth-Token: ABCDEFG....
```

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

4.3 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**管理员创建IAM用户**接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-Type”。

对于**管理员创建IAM用户**接口，返回如**图4-2**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 4-2 管理员创建 IAM 用户响应消息头

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

响应消息体

该部分可选。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**管理员创建IAM用户**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。


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


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


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


其中，error_code表示错误码，error_msg表示错误描述信息。

5 应用示例

5.1 创建 MRS 集群

场景描述

本章节指导用户通过API创建一个MRS分析集群。API的调用方法请参见**如何调用API**。

约束限制

- 已在待创建集群区域通过虚拟私有云服务创建VPC，子网。创建VPC请参考**查询VPC列表**和**创建VPC**。创建子网请参考**查询子网列表**和**创建子网**。
- 已获取待创建集群区域的区域和可用区信息，具体请参考**终端节点**。
- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已确定待创建集群的版本及版本支持的组件信息，请参见**MRS组件版本一览表**。
- 该示例创建出来的是按需购买的分析集群。

操作步骤

- 接口相关信息

  URI格式：POST /v2/{project_id}/clusters

  详情请参见**创建集群**。
- 请求示例

  POST: https://*{endpoint}*/v2/*{project_id}*/clusters
  - {endpoint}信息具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - 节点的实例规格参数“node_size”建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。

  Body：

```
{
 "cluster_version": "MRS 3.2.0-LTS.1",
 "cluster_name": "mrs_Demo",
 "cluster_type": "ANALYSIS",
 "charge_info": {
    "charge_mode": "postPaid"
```



```
},
"region": "",
"availability_zone": "",
"vpc_name": "vpc-37cd",
"subnet_name": "subnet-ed99",
"components": "Hadoop,Spark2x,HBase,Hive,Hue,Flink,Oozie,Ranger,Tez",
"safe_mode": "KERBEROS",
"manager_admin_password": "xxx",
"login_mode": "PASSWORD",
"node_root_password": "xxx",
"log_collection": 1,
"mrs_ecs_default_agency": "MRS_ECS_DEFAULT_AGENCY",
"tags": [
 {
   "key": "tag1",
   "value": "111"
 },
 {
   "key": "tag2",
   "value": "222"
 }
],
"node_groups": [
 {
   "group_name": "master_node_default_group",
   "node_num": 2,
   "node_size": "rc3.4xlarge.4.linux.bigdata",
   "root_volume": {
     "type": "SAS",
     "size": 480
   },
   "data_volume": {
     "type": "SAS",
     "size": 600
   },
   "data_volume_count": 1
 },
{
   "group_name": "core_node_analysis_group",
   "node_num": 3,
   "node_size": "rc3.4xlarge.4.linux.bigdata",
   "root_volume": {
     "type": "SAS",
     "size": 480
   },
   "data_volume": {
     "type": "SAS",
     "size": 600
   },
   "data_volume_count": 1
 },
 {
   "group_name": "task_node_analysis_group",
   "node_num": 3,
   "node_size": "rc3.4xlarge.4.linux.bigdata",
   "root_volume": {
     "type": "SAS",
     "size": 480
   },
   "data_volume": {
     "type": "SAS",
     "size": 600
   },
   "data_volume_count": 1,
  "auto_scaling_policy": {
            "auto_scaling_enable": true,
            "min_capacity": 0,
            "max_capacity": 1,
            "resources_plans": [],
```

```
            "exec_scripts": [],
            "rules": [
               {
                  "name": "default-expand-1",
                  "description": "",
                  "adjustment_type": "scale_out",
                  "cool_down_minutes": 5,
                  "scaling_adjustment": "1",
                  "trigger": {
                     "metric_id": 2003,
                     "metric_name": "StormSlotAvailablePercentage",
                     "metric_value": 100,
                     "comparison_operator_id": 2003,
                     "comparison_operator": "LTOE",
                     "evaluation_periods": "1"
                  }
               }
            ]
         }
   }
 ]
}
```

参数详细信息请参考**创建集群**获取。

- 响应示例

```
{
   "cluster_id": "da1592c2-bb7e-468d-9ac9-83246e95447a"
}
```

5.2 扩容集群

场景描述

创建集群后，扩容集群Core节点或者Task节点。MRS集群创建成功后不支持调整Master节点数量，即不支持扩缩容Master节点。API的调用方法请参见**如何调用API**。

约束限制

- 集群已创建成功并处于“运行中”。
- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已获取集群ID，即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。
- 该示例以扩容Core节点为例。

操作步骤

- 接口相关信息

  URI格式：PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

  详情请参见**调整集群节点**。

- 请求示例

  PUT: https://*{endpoint}*/v1.1/*{project_id}*/cluster_infos/*{cluster_id}*

  - {endpoint}信息具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - {cluster_id}信息即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

Body：

```
{
   "service_id": "",
   "plan_id": "",
   "parameters": {
      "order_id": "",
      "scale_type": "scale_out",
      "node_id": "node_orderadd",
      "node_group": "core_node_default_group",
      "instances": "1",
     "skip_bootstrap_scripts":false,
     "scale_without_start":false
   },
   "previous_values": {
      "plan_id": ""
   }
}
```

参数详细信息请参考**调整集群节点**获取。

- 响应示例

```
{
   "result": "succeeded"
}
```

5.3 缩容集群

场景描述

创建集群后，缩容集群Core节点或者Task节点。MRS集群创建成功后不支持调整Master节点数量，即不支持扩缩容Master节点。API的调用方法请参见**如何调用API**。

约束限制

- 集群已创建成功并处于“运行中”。
- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已获取集群ID，即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。
- 该示例以缩容Core节点为例。

操作步骤

- 接口相关信息

  URI格式：PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

  详情请参见**调整集群节点**。

- 请求示例

  PUT: https://*{endpoint}*/v1.1/*{project_id}*/cluster_infos/*{cluster_id}*

  - {endpoint}信息具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - {cluster_id}信息即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

  Body：

```
{
   "service_id": "",
   "plan_id": "",
```

```
    "parameters": {
        "order_id": "",
        "scale_type": "scale_in",
        "node_id": "node_orderadd",
        "node_group": "core_node_default_group",
        "instances": "1"


    },
    "previous_values": {
        "plan_id": ""
    }
}
```

参数详细信息请参考**调整集群节点**获取。

- 响应示例

```
{
    "result": "succeeded"
}
```

5.4 新增作业

场景描述

在MRS集群中新增并提交一个作业。API的调用方法请参见**如何调用API**。

约束限制

- 集群已创建成功并处于“运行中”。
- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已获取集群ID，即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。
- IAM用户已同步完成，可通过在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“单击同步”进行IAM用户同步。
- 作业相关程序和输入文件已存放在OBS中。
- 该示例以新增MapReduce作业为例。

操作步骤

- 接口相关信息

  URI格式：POST /v2/{project_id}/clusters/{cluster_id}/job-executions

  详情请参见**新增并执行作业**。

- 请求示例

  POST: https://*{endpoint}*/v2/*{project_id}*/clusters/*{cluster_id}*/job-executions

  - {endpoint}信息具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - {cluster_id}信息即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

  Body：

```
{
    "job_name":"MapReduceTest",
    "job_type":"MapReduce",
    "arguments":[
```

```
        "obs://obs-test/program/hadoop-mapreduce-examples-x.x.x.jar",
        "wordcount",
        "obs://obs-test/input/",
        "obs://obs-test/job/mapreduce/output"
    ],
    "properties":{
        "fs.obs.endpoint":"obs endpoint",
        "fs.obs.access.key":"xxx",
        "fs.obs.secret.key":"yyy"
    }
}
```

参数详细信息请参考**新增并执行作业**获取。

- 响应示例

```
{
  "job_submit_result":{
      "job_id":"44b37a20-ffe8-42b1-b42b-78a5978d7e40",
      "state":"COMPLETE"
  }
}
```

5.5 终止作业

场景描述

当作业提交后未执行完成时，手动终止作业。API的调用方法请参见**如何调用API**。

约束限制

- 集群已创建成功并处于“运行中”。
- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已获取集群ID，即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。
- 已获取作业ID，即作业提交成功后返回结果中的“job_id”或参考**获取作业ID**获取。
- IAM用户已同步完成，可通过在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“单击同步”进行IAM用户同步。
- 作业相关程序和输入文件已存放在OBS中。
- 该示例以新增MapReduce作业为例。

操作步骤

- 接口相关信息

  URI格式：POST /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/kill

  详情请参见**终止作业**。

- 请求示例

  POST: https://*{endpoint}*/v2/*{project_id}*/clusters/*{cluster_id}*/job-executions/*{job_execution_id}*/kill

  - {endpoint}信息具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - {cluster_id}信息即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

  - {job_execution_id}信息即作业提交成功后返回结果中的“job_id”或参考**获取作业ID**获取。

  Body：无

- 响应示例

  无

5.6 删除集群

场景描述

数据完成处理分析后或者集群运行异常无法提供服务时可删除集群服务。

处于如下状态的集群不允许删除：

- scaling-out：扩容中
- scaling-in：缩容中
- starting：启动中
- terminating：删除中
- terminated：已删除
- failed：失败

API的调用方法请参见**如何调用API**。

约束限制

- 已获取待创建集群区域的项目ID，请参考**获取项目ID**获取。
- 已获取集群ID，即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

操作步骤

- 接口相关信息

  URI格式：DELETE /v1.1/{project_id}/clusters/{cluster_id}

  详情请参见**删除集群**。

- 请求示例

  DELETE: https://*{endpoint}*/v1.1/*{project_id}*/clusters/*{cluster_id}*

  - {endpoint}具体请参考**终端节点**。
  - {project_id}信息请通过**获取项目ID**获取。
  - {cluster_id}信息即创建集群成功后返回结果中的“cluster_id” 或参考**获取集群ID**获取。

  Body：无

- 响应示例

```
{
    "result": "succeeded"
}
```

6 API V2

6.1 集群管理接口

6.1.1 创建集群

功能介绍

创建一个MRS集群。使用接口前，您需要先获取下的资源信息。

- 通过VPC创建或查询VPC、子网
- 通过ECS创建或查询密钥对
- 通过**终端节点**获取区域信息
- 参考**MRS服务支持的组件**获取MRS版本及对应版本支持的组件信息

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters

表 6-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-2 请求 Body 参数

参数	是否必选	参数类型	描述
is_dec_project	否	Boolean	**参数解释：** 说明是否为专属云的资源。 **约束限制：** 不涉及 **取值范围：** ● true：是专属云的资源。 ● false：不是专属云的资源。 **默认取值：** false
cluster_version	是	String	**参数解释：** 集群版本。例如：MRS 3.1.0。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_name	是	String	**参数解释：** 集群名称。 **约束限制：** 不涉及 **取值范围：** ● 不允许相同。 ● 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_type	是	String	**参数解释：** 集群类型。 **约束限制：** 不涉及 **取值范围：** ● ANALYSIS：分析集群 ● STREAMING：流式集群 ● MIXED：混合集群 ● CUSTOM：自定义集群，仅MRS 3.x版本支持。 **默认取值：** 不涉及
charge_info	否	ChargeInfo object	**参数解释：** 计费类型信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
region	是	String	**参数解释：** 集群所在区域信息，请参见**终端节点**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpc_name	是	String	**参数解释：** 子网所在VPC名称。通过VPC管理控制台获取名称： 1. 登录VPC管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 在“虚拟私有云”页面的列表中即可获取VPC名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_id	否	String	**参数解释：** 子网ID。通过VPC管理控制台获取子网ID： 1. 登录VPC管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网。 4. 单击对应子网名称，获取“网络ID”。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_name	是	String	**参数解释：** 子网名称。通过VPC管理控制台获取子网名称： 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网，获取子网名称。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。当仅填写“subnet_name”一个参数且VPC下存在同名子网时，创建集群时以VPC平台第一个名称的子网为准。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及
components	是	String	**参数解释：** 组件名称列表，用逗号分隔。支持的组件请参见**获取MRS集群信息**页面的“MRS服务支持的组件”内容。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
external_data sources	否	Array of **ClusterDataConnectorMap** objects	**参数解释：** 部署Hive和Ranger等组件时，可以关联数据连接，将元数据存储于关联的数据库。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
availability_zone	是	String	**参数解释：** 可用分区名称，不支持多AZ集群。可用分区信息请参见**终端节点**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
security_groups_id	否	String	**参数解释：** 集群安全组的ID。 • 当该ID为空时MRS后台会自动创建安全组，自动创建的安全组名称以mrs_{cluster_name}开头。 • 当该ID不为空时，表示使用固定安全组来创建集群，传入的ID必须是当前租户中包含的安全组ID。 • 支持多个安全组ID，以逗号分隔。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
auto_create_default_security_group	否	Boolean	**参数解释：** 是否要创建MRS集群默认安全组。 **约束限制：** 当指定该参数为true，则无论“security_groups_id”参数是否指定，都会为集群创建默认安全组。 **取值范围：** ● true：创建MRS集群默认安全组。 ● false：不创建MRS集群默认安全组。 **默认取值：** false
safe_mode	是	String	**参数解释：** MRS集群运行模式。 **约束限制：** 不涉及 **取值范围：** ● SIMPLE：普通集群，表示Kerberos认证关闭，用户可使用集群提供的所有功能。 ● KERBEROS：安全集群，表示Kerberos认证开启，普通用户无权限使用MRS集群的“文件管理”和“作业管理”功能，并且无法查看Hadoop、Spark的作业记录以及集群资源使用情况。如果需要使用集群更多功能，需要找Manager的管理员分配权限。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
manager_admin_password	是	String	**参数解释：** 配置Manager管理员用户的密码。 **约束限制：** 不涉及 **取值范围：** ● 密码长度应在8-26个字符之间。 ● 至少包含四种字符组合，如大写字母，小写字母，数字，特殊字符（!@$%^-_=+[{}]:,./?），但不能包含空格。 ● 不能与用户名或者倒序用户名相同。 **默认取值：** 不涉及
login_mode	是	String	**参数解释：** 节点登录方式。 **约束限制：** 不涉及 **取值范围：** ● PASSWORD：密码登录，选择此项时，node_root_password不能为空。 ● KEYPAIR：密钥对登录，选择此项时，node_keypair_name不能为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_root_password	否	String	**参数解释：** 配置访问集群节点的root密码。 **约束限制：** 不涉及 **取值范围：** 密码设置约束如下： ● 字符串类型，可输入的字符串长度为8-26。 ● 至少包含四种字符组合，如大写字母，小写字母，数字，特殊字符（!@$%^-_=+[{}]:,./?），但不能包含空格。 ● 不能与用户名或者倒序用户名相同。 **默认取值：** 不涉及
node_keypair_name	否	String	**参数解释：** 密钥对名称。用户可以使用密钥对方式登录集群节点。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
enterprise_project_id	否	String	**参数解释：** 企业项目ID。创建集群时，给集群绑定企业项目ID。获取方式请参见《企业管理API参考》的“查询企业项目列表”响应消息表“enterprise_project字段数据结构说明”的“id”,即**表5 enterprise_project_list字段数据结构说明**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 默认设置为0，表示为default企业项目。
eip_address	否	String	**参数解释：** 与MRS集群绑定的弹性公网IP，可实现使用弹性公网IP访问Manager的目的。该弹性公网IP必须已经创建且与集群在同一区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eip_id	否	String	**参数解释：** 当“eip_address”配置时，该参数必须配置，用于表示绑定的弹性公网IP的ID。可通过在VPC服务的“网络控制台 > 弹性公网IP和带宽 > 弹性公网IP”页面单击待绑定的弹性公网IP，在基本信息中获取“ID”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
mrs_ecs_default_agency	否	String	**参数解释：** 集群节点默认绑定的委托名称，固定为MRS_ECS_DEFAULT_AGENCY。通过绑定委托，您可以将部分资源共享给ECS或BMS云服务来管理，例如通过配置ECS委托可自动获取AK/SK访问OBS。MRS_ECS_DEFAULT_AGENCY委托拥有对象存储服务的OBS OperateAccess权限和在集群所在区域拥有CES FullAccess（对开启细粒度策略的用户）、CES Administrator和KMS Administrator权限。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
template_id	否	String	**参数解释：** 当集群类型为CUSTOM时，用于指定节点部署所使用的模板。 ● mgmt_control_combined_v2：管控合设模板，管理角色和控制角色共同部署在Master节点中，数据实例合设在同一节点组。该部署方式适用于100个以下的节点，可以减少成本。 ● mgmt_control_separated_v2：管控分设模板，管理角色和控制角色分别部署在不同的Master节点中，数据实例合设在同一节点组。该部署方式适用于100-500个节点，在高并发负载情况下表现更好。 ● mgmt_control_data_separated_v2：数据分设模板，管理角色和控制角色分别部署在不同的Master节点中，数据实例分设在不同节点组。该部署方式适用于500个以上的节点，可以将各组件进一步分开部署，适用于更大的集群规模。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	否	Array of **Tag** objects	**参数解释：** 集群的标签信息。 **约束限制：** 同一个集群最多能使用10个tag，tag的名称（key）不能重复。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
log_collection	否	Integer	**参数解释：** 集群创建失败时，是否收集失败日志。 **约束限制：** 不涉及 **取值范围：** ● 0：不创建OBS桶仅用于MRS集群创建失败时的日志收集。 ● 1：创建OBS桶仅用于MRS集群创建失败时的日志收集。 **默认取值：** 1
node_groups	是	Array of **NodeGroupV2** objects	**参数解释：** 组成集群的节点组信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
bootstrap_scripts	否	Array of **BootstrapScript** objects	**参数解释：** 配置引导操作脚本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
add_jobs	否	Array of **AddJobsReqV11** objects	**参数解释：** 创建集群时可同时提交作业，当前仅MRS1.8.7之前版本支持，暂时只支持新增一个作业。建议使用创建集群并提交作业接口RunJobFlow的steps参数。 **约束限制：** 不能超过1条。 **取值范围：** 不涉及 **默认取值：** 不涉及
log_uri	否	String	**参数解释：** 集群日志转储至OBS的具体路径。开启日志转储功能后，日志上传需要对应OBS路径的读写权限，请配置MRS_ECS_DEFULT_AGENCY默认委托或具有对应OBS路径读写权限的自定义委托。具体请参见**配置存算分离集群（委托方式）**。该参数只适用于支持“集群日志转储OBS”特性的集群版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
component_configs	否	Array of **ComponentConfig** objects	**参数解释：** 集群组件自定义配置。该参数只适用于支持“自定义组件配置创建集群”特性的集群版本。 **约束限制：** 不能超过50条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
smn_notify	否	SmnNotify object	**参数解释：** smn告警订阅。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-3 ClusterDataConnectorMap

参数	是否必选	参数类型	描述
map_id	否	Integer	**参数解释：** 数据连接关联ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
connector_id	否	String	**参数解释：** 数据连接ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
component_name	否	String	**参数解释：** 组件名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
role_type	否	String	**参数解释：** 组件角色类型。 **约束限制：** 不涉及 **取值范围：** • hive_metastore：Hive Metastore角色 • hive_data：Hive角色 • hbase_data：Hbase角色 • ranger_data：Ranger角色 **默认取值：** 不涉及
source_type	否	String	**参数解释：** 数据连接类型。 **约束限制：** 不涉及 **取值范围：** • LOCAL_DB：本地元数据 • RDS_POSTGRES：RDS服务PostgreSQL数据库 • RDS_MYSQL：RDS服务MySQL数据库 • gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及
cluster_id	否	String	**参数解释：** 关联集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
status	否	Integer	**参数解释：** 数据连接状态。 **约束限制：** 不涉及 **取值范围：** ● 0：代表正常状态 ● 1：代表使用中 **默认取值：** 不涉及

表 6-4 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 同一资源的key值不能重复。 ● 最大长度128个unicode字符，不能为空字符串。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 最大长度255个unicode字符，可以为空字符串。 **默认取值：** 不涉及

表 6-5 NodeGroupV2

参数	是否必选	参数类型	描述
group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”组成，且长度为[1-64]个字符。 节点组配置原则如下： ● master_node_default_group：Master节点组，所有集群类型均需包含该节点组。 ● core_node_analysis_group：分析Core节点组，分析集群、混合集群均需包含该节点组。 ● core_node_streaming_group：流式Core节点组，流式集群和混合集群均需包含该节点组。 ● task_node_analysis_group：分析Task节点组，分析集群和混合集群可根据需要选择该节点组。 ● task_node_streaming_group：流式Task节点组，流式集群、混合集群可根据需要选择该节点组。 ● node_group{x}：自定义集群节点组，可根据需要添加多个，最多支持添加9个该节点组。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_num	是	Integer	**参数解释：** 节点数量。 **约束限制：** Core与Task节点总数最大为500个。 **取值范围：** 0-500 **默认取值：** 不涉及
node_size	是	String	**参数解释：** 节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume	否	**Volume** object	**参数解释：** 节点系统盘信息，部分虚拟机或BMS自带系统盘的情况该参数可选，其他情况该参数必选。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_volume	否	**Volume** object	**参数解释：** 节点数据盘信息。 **约束限制：** 当data_volume_count不为0时，该参数必选。 **取值范围：** 不涉及 **默认取值：** 不涉及
data_volume_count	否	Integer	**参数解释：** 节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 0-20 **默认取值：** 不涉及
charge_info	否	**ChargeInfo** object	**参数解释：** 节点组的计费类型，Master和Core节点组是和集群的计费类型一致，Task节点组可以和集群的计费类型不同。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
auto_scaling_policy	否	**AutoScalingPolicy** object	**参数解释：** 弹性伸缩规则信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
assigned_roles	否	Array of strings	**参数解释：** 当集群类型为CUSTOM时，该参数必选。可以指定节点组中部署的角色，该参数是一个字符串数组，每个字符串表示一个角色表达式。 角色表达式定义： ● 当该角色在节点组所有节点部署时： {role name}，如“DataNode”。 ● 当该角色在节点组指定下标节点部署时：{role name}:{index1},{index2}…,{indexN}，如“NameNode:1,2”，下标从1开始计数。 可选的角色请参考**MRS支持的角色与组件对应表**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-6 Volume

参数	是否必选	参数类型	描述
type	是	String	**参数解释：** 磁盘类型。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO磁盘类型。 ● SAS：高IO磁盘类型。 ● SSD：超高IO磁盘类型。 ● GPSSD：通用型SSD磁盘类型 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
size	是	Integer	**参数解释：** 数据盘大小，容量单位为GB。 **约束限制：** 不涉及 **取值范围：** 10-32768 **默认取值：** 不涉及

表 6-7 ChargeInfo

参数	是否必选	参数类型	描述
charge_mode	是	String	**参数解释：** 计费模式。 **约束限制：** 不涉及 **取值范围：** ● prePaid：预付费，即包年/包月。（创建集群接口现已支持预付费，创建集群并提交作业接口暂不支持预付费。） ● postPaid：后付费，即按需计费。 **默认取值：** 不涉及
period_type	否	String	**参数解释：** 周期类型。 **约束限制：** 不涉及 **取值范围：** ● month：包月。 ● year： 包年。 ● day：按需计费。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
period_num	否	Integer	**参数解释：** 周期数。 **约束限制：** “charge_mode”为“prePaid”时生效，且为必选值，指定订购的时间。 **取值范围：** ● 当“period_type”为“month”时，取值为1-9。 ● 当“period_type”为“year”时，取值为1-3。 **默认取值：** 不涉及
is_auto_pay	否	Boolean	**参数解释：** 是否自动支付，包周期模式下使用，下单订购后，是否自动从客户的账户中支付，而不需要客户手动去进行支付。 **约束限制：** 不涉及 **取值范围：** ● true：自动支付，会自动选择折扣和优惠券进行优惠，然后自动从客户账户中支付，自动支付失败后会生成订单成功、但订单状态为“待支付”，等待客户手动支付。 ● false：手动支付，需要客户手动去支付，客户可以选择折扣和优惠券。 **默认取值：** false

表 6-8 AutoScalingPolicy

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** ● true：开启自动伸缩规则 ● false：不开启自动伸缩规则 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
exec_scripts	否	Array of **ScaleScript** objects	**参数解释：** 弹性伸缩自定义自动化脚本列表。若该参数为空表示不启用自动化脚本。在V2弹性伸缩策略创建和更新接口中暂时不支持该字段。 **约束限制：** 不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-9 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	**参数解释：** 资源计划的起始时间，格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
effective_days	否	Array of strings	**参数解释：** 资源计划的生效日期，为空时代表每日，另外也可为以下返回值： MONDAY（周一）、TUESDAY（周二）、WEDNESDAY（周三）、THURSDAY（周四）、FRIDAY（周五）、SATURDAY（周六）、SUNDAY（周日） **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-10 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 在一个节点组范围内，不允许重名。 **默认取值：** 不涉及
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度为[0-1024]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** • scale_out：扩容 • scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-11 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。详细指标名称内容请参见**"弹性伸缩指标列表"**。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见**"弹性伸缩指标列表"**。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** • LT：小于 • GT：大于 • LTOE：小于等于 • GTOE：大于等于 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-288 **默认取值：** 不涉及

表 6-12 ScaleScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩自定义自动化脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的自定义自动化脚本名称不允许相同。 只能由英文字母、数字、空格以及“_”和“-”组成，不能以空格开头，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 自定义自动化脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。示例：obs://XXX/scale.sh ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 自定义自动化脚本参数。多个参数间用空格隔开。 可以传入以下系统预定义参数： ● ${mrs_scale_node_num}：扩缩容节点数 ● ${mrs_scale_type}：扩缩容类型，扩容为scale_out，缩容为scale_in ● ${mrs_scale_node_hostnames}：扩缩容的节点主机名称 ● ${mrs_scale_node_ips}：扩缩容的节点IP ● ${mrs_scale_rule_name}：触发扩缩容的规则名 其他用户自定义参数使用方式与普通shell脚本相同，多个参数中间用空格隔开。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 自定义自动化脚本所执行的节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 自定义自动化脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：自定义自动化脚本只运行在主Master节点上。 ● false：自定义自动化脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	是	String	**参数解释：** 自定义自动化脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此自定义自动化脚本是否执行成功，则集群都能继续安装和启动。 **约束限制：** 由于缩容成功无法回滚，因此缩容后执行的脚本“fail_action”必须设置为“continue”。 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
action_stage	是	String	**参数解释：** 脚本执行时机。 **约束限制：** 不涉及 **取值范围：** ● before_scale_out：扩容前 ● before_scale_in：缩容前 ● after_scale_out：扩容后 ● after_scale_in：缩容后 **默认取值：** 不涉及

表 6-13 BootstrapScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 引导操作脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的引导操作脚本名称不允许相同。 只能由英文字母、数字、空格以及“_”和“-”组成，不能以空格开头，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 引导操作脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 OBS桶的路径：直接手动输入脚本路径。例如输入MRS提供的公共样例脚本路径。示例：obs://bootstrap/presto/presto-install.sh，其中安装dualroles时，presto-install.sh脚本参数为dualroles, 安装worker时，presto-install.sh脚本参数为worker。根据Presto使用习惯，建议您在Active Master节点上安装dualroles，在Core节点上安装worker。 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 引导操作脚本参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 引导操作脚本所执行的节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 引导操作脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本只运行在主Master节点上。 ● false：引导操作脚本可运行在所有Master节点上。 **默认取值：** 不涉及
fail_action	是	String	**参数解释：** 引导操作脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“继续”，无论此引导操作是否执行成功，则集群都能继续安装和启动。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
before_component_start	否	Boolean	**参数解释：** 引导操作脚本执行的时间。目前支持“组件启动前”和“组件启动后”两种类型。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本在组件启动前执行。 ● false：引导操作脚本在组件启动后执行。 **默认取值：** false
start_time	否	Long	**参数解释：** 单个引导操作脚本的执行时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
state	否	String	**参数解释：** 单个引导操作脚本的运行状态。 **约束限制：** 不涉及 **取值范围：** ● PENDING：挂起 ● IN_PROGRESS：处理中 ● SUCCESS：处理成功 ● FAILURE：处理失败 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
action_stages	否	Array of strings	**参数解释：** 选择引导操作脚本执行的时间。 **约束限制：** 参数枚举值： ● BEFORE_COMPONENT_FIRST_START: 组件首次启动前 ● AFTER_COMPONENT_FIRST_START: 组件首次启动后 ● BEFORE_SCALE_IN: 缩容前 ● AFTER_SCALE_IN: 缩容后 ● BEFORE_SCALE_OUT: 扩容前 ● AFTER_SCALE_OUT: 扩容后 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-14 AddJobsReqV11

参数	是否必选	参数类型	描述
job_type	是	Integer	**参数解释：** 作业类型码。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● Spark ● Hive Script ● HiveSQL（当前不支持） ● DistCp，导入、导出数据，（当前不支持）。 ● Spark Script ● Spark SQL，提交SQL语句，（当前不支持）。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_name	是	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及
jar_path	否	String	**参数解释：** 执行程序Jar包或sql文件地址。 **约束限制：** 不涉及 **取值范围：** ● 最多为1023字符，不能包含;|&>,<'$特殊字符，且不可为空或全空格。 ● 文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 – OBS：以“obs://”开头。不支持KMS加密的文件或程序。 – HDFS：以“/”开头。 ● Spark Script需要以“.sql”结尾，MapReduce和Spark Jar需要以“.jar”结尾，sql和jar不区分大小写。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
arguments	否	String	**参数解释：** 程序执行的关键参数，该参数由用户程序内的函数指定，MRS只负责参数的传入。 **约束限制：** 不涉及 **取值范围：** 最多为150000字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
input	否	String	**参数解释：** 数据输入地址。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 ● OBS：以“obs://”开头。不支持KMS加密的文件或程序。 ● HDFS：以“/”开头。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
output	否	String	**参数解释：** 数据输出地址。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。如果该路径不存在，系统会自动创建。 ● OBS：以“obs://”开头。 ● HDFS：以“/”开头。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_log	否	String	**参数解释：** 作业日志存储地址，该日志信息记录作业运行状态。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 ● OBS：以“obs://”开头。 ● HDFS：以“/”开头。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
hive_script_path	否	String	**参数解释：** sql程序路径，仅Spark Script和Hive Script作业需要使用此参数。 **约束限制：** 不涉及 **取值范围：** ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 – OBS：以“obs://”开头。不支持KMS加密的文件或程序。 – HDFS：以“/”开头。 ● 需要以“.sql”结尾，sql不区分大小写。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
hql	否	String	**参数解释：** HQL脚本语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
shutdown_cluster	否	Boolean	**参数解释：** 作业执行完成后，是否删除集群。 **约束限制：** 不涉及 **取值范围：** ● true：删除集群 ● false：不删除集群 **默认取值：** 不涉及
submit_job_once_cluster_run	是	Boolean	**参数解释：** 创建集群时是否同时提交作业。此处应设置为true。 **约束限制：** 不涉及 **取值范围：** ● true：创建集群同时提交作业 ● false：单独提交作业 **默认取值：** 不涉及
file_action	否	String	**参数解释：** 数据导入导出。 **约束限制：** 不涉及 **取值范围：** ● import：导入数据 ● export：导出数据 **默认取值：** 不涉及

表 6-15 ComponentConfig

参数	是否必选	参数类型	描述
component_name	是	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
configs	否	Array of **Config** objects	**参数解释：** 组件配置项列表。 **约束限制：** 不能超过100条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-16 Config

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 配置名，仅支持MRS组件配置页面上所展示的配置名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 配置值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
config_file_name	是	String	**参数解释：** 配置文件名，仅支持MRS组件配置页面上所展示的文件名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-17 SmnNotify

参数	是否必选	参数类型	描述
topic_urn	否	String	**参数解释：** SMN消息通知服务的主题urn。 **约束限制：** 如果需要开启告警订阅，则必填。 **取值范围：** 不涉及 **默认取值：** 不涉及
subscription_name	否	String	**参数解释：** 该订阅规则名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** default_alert_rule

响应参数

状态码： 200

表 6-18 响应 Body 参数

参数	参数类型	描述
cluster_id	String	**参数解释：** 集群创建成功后系统返回的集群ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- 创建一个分析集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3；一个Task节点组，节点数为3。每周一的12点至13点开启弹性伸缩。Hive组件的初始配置hive.union.data.type.incompatible.enable修改为true，dfs.replication修改为4。

```
POST /v2/{project_id}/clusters

{
  "cluster_version" : "MRS 3.1.0",
  "cluster_name" : "mrs_DyJA_dm",
  "cluster_type" : "ANALYSIS",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "region" : "",
  "availability_zone" : "",
  "vpc_name" : "vpc-37cd",
  "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
  "subnet_name" : "subnet",
  "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,Ranger,Tez",
  "safe_mode" : "KERBEROS",
  "manager_admin_password" : "your password",
  "login_mode" : "PASSWORD",
  "node_root_password" : "your password",
  "log_collection" : 1,
  "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
  "tags" : [ {
    "key" : "tag1",
    "value" : "111"
  }, {
    "key" : "tag2",
    "value" : "222"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 2,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
```

```
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "core_node_analysis_group",
  "node_num" : 3,
  "node_size" : "rc3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "task_node_analysis_group",
  "node_num" : 3,
  "node_size" : "rc3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1,
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 0,
    "max_capacity" : 1,
    "resources_plans" : [ {
      "period_type" : "daily",
      "start_time" : "12:00",
      "end_time" : "13:00",
      "min_capacity" : 2,
      "max_capacity" : 3,
      "effective_days" : [ "MONDAY" ]
    } ],
    "exec_scripts" : [ {
      "name" : "test",
      "uri" : "s3a://obs-mrstest/bootstrap/basic_success.sh",
      "parameters" : "",
      "nodes" : [ "master_node_default_group", "core_node_analysis_group",
"task_node_analysis_group" ],
      "active_master" : false,
      "action_stage" : "before_scale_out",
      "fail_action" : "continue"
    } ],
    "rules" : [ {
      "name" : "default-expand-1",
      "description" : "",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : "1",
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : 100,
        "comparison_operator" : "GTOE",
        "evaluation_periods" : "1"
      }
```

```
    } ]
   }
  } ],
  "component_configs" : [ {
   "component_name" : "Hive",
   "configs" : [ {
    "key" : "hive.union.data.type.incompatible.enable",
    "value" : "true",
    "config_file_name" : "hive-site.xml"
   }, {
    "key" : "dfs.replication",
    "value" : "4",
    "config_file_name" : "hdfs-site.xml"
   } ]
  } ]
}
```

- 创建一个流式集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3，一个Task节点组，节点数为0。每周一的12点至13点开启弹性伸缩。

```
POST /v2/{project_id}/clusters

{
 "cluster_version" : "MRS 3.1.0",
 "cluster_name" : "mrs_Dokle_dm",
 "cluster_type" : "STREAMING",
 "charge_info" : {
  "charge_mode" : "postPaid"
 },
 "region" : "",
 "availability_zone" : "",
 "vpc_name" : "vpc-37cd",
 "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
 "subnet_name" : "subnet",
 "components" : "Storm,Kafka,Flume,Ranger",
 "safe_mode" : "KERBEROS",
 "manager_admin_password" : "your password",
 "login_mode" : "PASSWORD",
 "node_root_password" : "your password",
 "log_collection" : 1,
 "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
 "tags" : [ {
  "key" : "tag1",
  "value" : "111"
 }, {
  "key" : "tag2",
  "value" : "222"
 } ],
 "node_groups" : [ {
  "group_name" : "master_node_default_group",
  "node_num" : 2,
  "node_size" : "rc3.4xlarge.4.linux.bigdata",
  "root_volume" : {
   "type" : "SAS",
   "size" : 480
  },
  "data_volume" : {
   "type" : "SAS",
   "size" : 600
  },
  "data_volume_count" : 1
 }, {
  "group_name" : "core_node_streaming_group",
  "node_num" : 3,
  "node_size" : "rc3.4xlarge.4.linux.bigdata",
  "root_volume" : {
   "type" : "SAS",
   "size" : 480
  },
```

```
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
 }, {
  "group_name" : "task_node_streaming_group",
  "node_num" : 0,
  "node_size" : "rc3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1,
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 0,
    "max_capacity" : 1,
    "resources_plans" : [ {
      "period_type" : "daily",
      "start_time" : "12:00",
      "end_time" : "13:00",
      "min_capacity" : 2,
      "max_capacity" : 3,
      "effective_days" : [ "MONDAY" ]
    } ],
    "rules" : [ {
      "name" : "default-expand-1",
      "description" : "",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : "1",
      "trigger" : {
        "metric_name" : "StormSlotAvailablePercentage",
        "metric_value" : 100,
        "comparison_operator" : "LTOE",
        "evaluation_periods" : "1"
      }
    } ]
  }
 } ]
}
```

- 创建一个混合集群，集群版本号为MRS 3.1.0。其中包含一个Master节点组，节点数为2；两个Core节点组，每个Core节点组的节点数均为3；两个Task节点组，一个Task节点组节点数为1，另一个节点数为0。

```
POST /v2/{project_id}/clusters

{
 "cluster_version" : "MRS 3.1.0",
 "cluster_name" : "mrs_onmm_dm",
 "cluster_type" : "MIXED",
 "charge_info" : {
   "charge_mode" : "postPaid"
 },
 "region" : "",
 "availability_zone" : "",
 "vpc_name" : "vpc-37cd",
 "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
 "subnet_name" : "subnet",
 "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
 "safe_mode" : "KERBEROS",
 "manager_admin_password" : "your password",
 "login_mode" : "PASSWORD",
 "node_root_password" : "your password",
```

```
"log_collection" : 1,
"mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
"tags" : [ {
  "key" : "tag1",
  "value" : "111"
}, {
  "key" : "tag2",
  "value" : "222"
} ],
"node_groups" : [ {
  "group_name" : "master_node_default_group",
  "node_num" : 2,
  "node_size" : "Sit3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "core_node_streaming_group",
  "node_num" : 3,
  "node_size" : "Sit3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "core_node_analysis_group",
  "node_num" : 3,
  "node_size" : "Sit3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "task_node_analysis_group",
  "node_num" : 1,
  "node_size" : "Sit3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
  "data_volume" : {
    "type" : "SAS",
    "size" : 600
  },
  "data_volume_count" : 1
}, {
  "group_name" : "task_node_streaming_group",
  "node_num" : 0,
  "node_size" : "Sit3.4xlarge.4.linux.bigdata",
  "root_volume" : {
    "type" : "SAS",
    "size" : 480
  },
```



```
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1
  } ]
}
```

- 创建自定义管控合设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为3；两个Core节点组，一个节点数为3，另一个节点数为1。

```
POST /v2/{project_id}/clusters

{
  "cluster_version" : "MRS 3.1.0",
  "cluster_name" : "mrs_heshe_dm",
  "cluster_type" : "CUSTOM",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "region" : "",
  "availability_zone" : "",
  "vpc_name" : "vpc-37cd",
  "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
  "subnet_name" : "subnet",
  "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
  "safe_mode" : "KERBEROS",
  "manager_admin_password" : "your password",
  "login_mode" : "PASSWORD",
  "node_root_password" : "your password",
  "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
  "template_id" : "mgmt_control_combined_v2",
  "log_collection" : 1,
  "tags" : [ {
    "key" : "tag1",
    "value" : "111"
  }, {
    "key" : "tag2",
    "value" : "222"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 3,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "OMSServer:1,2", "SlapdServer:1,2", "KerberosServer:1,2", "KerberosAdmin:1,2",
"quorumpeer:1,2,3", "NameNode:2,3", "Zkfc:2,3", "JournalNode:1,2,3", "ResourceManager:2,3",
"JobHistoryServer:2,3", "DBServer:1,3", "Hue:1,3", "LoaderServer:1,3", "MetaStore:1,2,3",
"WebHCat:1,2,3", "HiveServer:1,2,3", "HMaster:2,3", "MonitorServer:1,2", "Nimbus:1,2", "UI:1,2",
"JDBCServer2x:1,2,3", "JobHistory2x:2,3", "SparkResource2x:1,2,3", "oozie:2,3", "LoadBalancer:2,3",
"TezUI:1,3", "TimelineServer:3", "RangerAdmin:1,2", "UserSync:2", "TagSync:2", "KerberosClient",
"SlapdClient", "meta", "HSConsole:2,3", "FlinkResource:1,2,3", "DataNode:1,2,3",
"NodeManager:1,2,3", "IndexServer2x:1,2", "ThriftServer:1,2,3", "RegionServer:1,2,3",
"ThriftServer1:1,2,3", "RESTServer:1,2,3", "Broker:1,2,3", "Supervisor:1,2,3", "Logviewer:1,2,3",
"Flume:1,2,3", "HSBroker:1,2,3" ]
  }, {
    "group_name" : "node_group_1",
    "node_num" : 3,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
```

```
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "DataNode", "NodeManager", "RegionServer", "Flume:1", "Broker",
"Supervisor", "Logviewer", "HBaseIndexer", "KerberosClient", "SlapdClient", "meta", "HSBroker:1,2",
"ThriftServer", "ThriftServer1", "RESTServer", "FlinkResource" ]
  }, {
    "group_name" : "node_group_2",
    "node_num" : 1,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "NodeManager", "KerberosClient", "SlapdClient", "meta", "FlinkResource" ]
  } ]
}
```

- 创建自定义管控分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为5；一个Core节点组，节点数为3。

```
POST /v2/{project_id}/clusters

{
  "cluster_version" : "MRS 3.1.0",
  "cluster_name" : "mrs_jdRU_dm01",
  "cluster_type" : "CUSTOM",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "region" : "",
  "availability_zone" : "",
  "vpc_name" : "vpc-37cd",
  "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
  "subnet_name" : "subnet",
  "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
  "safe_mode" : "KERBEROS",
  "manager_admin_password" : "your password",
  "login_mode" : "PASSWORD",
  "node_root_password" : "your password",
  "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
  "log_collection" : 1,
  "template_id" : "mgmt_control_separated_v2",
  "tags" : [ {
    "key" : "aaa",
    "value" : "111"
  }, {
    "key" : "bbb",
    "value" : "222"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 5,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
```

```
    "data_volume_count" : 1,
    "assigned_roles" : [ "OMSServer:1,2", "SlapdServer:3,4", "KerberosServer:3,4", "KerberosAdmin:3,4",
"quorumpeer:3,4,5", "NameNode:4,5", "Zkfc:4,5", "JournalNode:1,2,3,4,5", "ResourceManager:4,5",
"JobHistoryServer:4,5", "DBServer:3,5", "Hue:1,2", "LoaderServer:1,2", "MetaStore:1,2,3,4,5",
"WebHCat:1,2,3,4,5", "HiveServer:1,2,3,4,5", "HMaster:4,5", "MonitorServer:1,2", "Nimbus:1,2",
"UI:1,2", "JDBCServer2x:1,2,3,4,5", "JobHistory2x:4,5", "SparkResource2x:1,2,3,4,5", "oozie:1,2",
"LoadBalancer:1,2", "TezUI:1,2", "TimelineServer:5", "RangerAdmin:1,2", "KerberosClient",
"SlapdClient", "meta", "HSConsole:1,2", "FlinkResource:1,2,3,4,5", "DataNode:1,2,3,4,5",
"NodeManager:1,2,3,4,5", "IndexServer2x:1,2", "ThriftServer:1,2,3,4,5", "RegionServer:1,2,3,4,5",
"ThriftServer1:1,2,3,4,5", "RESTServer:1,2,3,4,5", "Broker:1,2,3,4,5", "Supervisor:1,2,3,4,5",
"Logviewer:1,2,3,4,5", "Flume:1,2,3,4,5", "HBaseIndexer:1,2,3,4,5", "TagSync:1", "UserSync:1" ]
  }, {
    "group_name" : "node_group_1",
    "node_num" : 3,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "DataNode", "NodeManager", "RegionServer", "Flume:1", "Broker",
"Supervisor", "Logviewer", "HBaseIndexer", "KerberosClient", "SlapdClient", "meta", "HSBroker:1,2",
"ThriftServer", "ThriftServer1", "RESTServer", "FlinkResource" ]
  } ]
}
```

- 创建自定义数据分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为9；四个Core节点组，每个Core节点组的节点数均为3。

```
POST /v2/{project_id}/clusters

{
  "cluster_version" : "MRS 3.1.0",
  "cluster_name" : "mrs_jdRU_dm02",
  "cluster_type" : "CUSTOM",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "region" : "",
  "availability_zone" : "",
  "vpc_name" : "vpc-37cd",
  "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
  "subnet_name" : "subnet",
  "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
  "safe_mode" : "KERBEROS",
  "manager_admin_password" : "your password",
  "login_mode" : "PASSWORD",
  "node_root_password" : "your password",
  "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
  "template_id" : "mgmt_control_data_separated_v2",
  "log_collection" : 1,
  "tags" : [ {
    "key" : "aaa",
    "value" : "111"
  }, {
    "key" : "bbb",
    "value" : "222"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 9,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
```

```
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "OMSServer:1,2", "SlapdServer:5,6", "KerberosServer:5,6", "KerberosAdmin:5,6",
"quorumpeer:5,6,7,8,9", "NameNode:3,4", "Zkfc:3,4", "JournalNode:5,6,7", "ResourceManager:8,9",
"JobHistoryServer:8", "DBServer:8,9", "Hue:8,9", "FlinkResource:3,4", "LoaderServer:3,5",
"MetaStore:8,9", "WebHCat:5", "HiveServer:8,9", "HMaster:8,9", "FTP-Server:3,4", "MonitorServer:3,4",
"Nimbus:8,9", "UI:8,9", "JDBCServer2x:8,9", "JobHistory2x:8,9", "SparkResource2x:5,6,7", "oozie:4,5",
"EsMaster:7,8,9", "LoadBalancer:8,9", "TezUI:5,6", "TimelineServer:5", "RangerAdmin:4,5",
"UserSync:5", "TagSync:5", "KerberosClient", "SlapdClient", "meta", "HSBroker:5", "HSConsole:3,4",
"FlinkResource:3,4" ]
  }, {
    "group_name" : "node_group_1",
    "node_num" : 3,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "DataNode", "NodeManager", "RegionServer", "Flume:1", "GraphServer",
"KerberosClient", "SlapdClient", "meta", "HSBroker:1,2" ]
  }, {
    "group_name" : "node_group_2",
    "node_num" : 3,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "HBaseIndexer", "SolrServer[3]", "EsNode[2]", "KerberosClient", "SlapdClient",
"meta", "SolrServerAdmin:1,2" ]
  }, {
    "group_name" : "node_group_3",
    "node_num" : 3,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "KerberosClient", "SlapdClient", "meta" ]
  }, {
    "group_name" : "node_group_4",
    "node_num" : 3,
    "node_size" : "rc3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
```



```
    "data_volume_count" : 1,
    "assigned_roles" : [ "Broker", "Supervisor", "Logviewer", "KerberosClient", "SlapdClient", "meta" ]
  } ]
}
```

响应示例

状态码： 200

正常响应示例。

```
{
  "cluster_id" : "da1592c2-bb7e-468d-9ac9-83246e95447a"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建一个分析集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3；一个Task节点组，节点数为3。每周一的12点至13点开启弹性伸缩。Hive组件的初始配置hive.union.data.type.incompatible.enable修改为true，dfs.replication修改为4。

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

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

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        List<Config> listComponentConfigsConfigs = new ArrayList<>();
        listComponentConfigsConfigs.add(
```

```
        new Config()
            .withKey("hive.union.data.type.incompatible.enable")
            .withValue("true")
            .withConfigFileName("hive-site.xml")
    );
    listComponentConfigsConfigs.add(
        new Config()
            .withKey("dfs.replication")
            .withValue("4")
            .withConfigFileName("hdfs-site.xml")
    );
    List<ComponentConfig> listbodyComponentConfigs = new ArrayList<>();
    listbodyComponentConfigs.add(
        new ComponentConfig()
            .withComponentName("Hive")
            .withConfigs(listComponentConfigsConfigs)
    );
    List<String> listExecScriptsNodes = new ArrayList<>();
    listExecScriptsNodes.add("master_node_default_group");
    listExecScriptsNodes.add("core_node_analysis_group");
    listExecScriptsNodes.add("task_node_analysis_group");
    List<ScaleScript> listAutoScalingPolicyExecScripts = new ArrayList<>();
    listAutoScalingPolicyExecScripts.add(
        new ScaleScript()
            .withName("test")
            .withUri("s3a://obs-mrstest/bootstrap/basic_success.sh")
            .withParameters("")
            .withNodes(listExecScriptsNodes)
            .withActiveMaster(false)
            .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
            .withActionStage(ScaleScript.ActionStageEnum.fromValue("before_scale_out"))
    );
    Trigger triggerRules = new Trigger();
    triggerRules.withMetricName("YARNAppRunning")
        .withMetricValue("100")
        .withComparisonOperator("GTOE")
        .withEvaluationPeriods(1);
    List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
    listAutoScalingPolicyRules.add(
        new Rule()
            .withName("default-expand-1")
            .withDescription("")
            .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
            .withCoolDownMinutes(5)
            .withScalingAdjustment(1)
            .withTrigger(triggerRules)
    );
    List<ResourcesPlan.EffectiveDaysEnum> listResourcesPlansEffectiveDays = new ArrayList<>();
    listResourcesPlansEffectiveDays.add(ResourcesPlan.EffectiveDaysEnum.fromValue("MONDAY"));
    List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
    listAutoScalingPolicyResourcesPlans.add(
        new ResourcesPlan()
            .withPeriodType("daily")
            .withStartTime("12:00")
            .withEndTime("13:00")
            .withMinCapacity(2)
            .withMaxCapacity(3)
            .withEffectiveDays(listResourcesPlansEffectiveDays)
    );
    AutoScalingPolicy autoScalingPolicyNodeGroups = new AutoScalingPolicy();
    autoScalingPolicyNodeGroups.withAutoScalingEnable(true)
        .withMinCapacity(0)
        .withMaxCapacity(1)
        .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
        .withRules(listAutoScalingPolicyRules)
        .withExecScripts(listAutoScalingPolicyExecScripts);
    Volume dataVolumeNodeGroups = new Volume();
    dataVolumeNodeGroups.withType("SAS")
        .withSize(600);
```

```
Volume rootVolumeNodeGroups = new Volume();
rootVolumeNodeGroups.withType("SAS")
    .withSize(480);
Volume dataVolumeNodeGroups1 = new Volume();
dataVolumeNodeGroups1.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups1 = new Volume();
rootVolumeNodeGroups1.withType("SAS")
    .withSize(480);
Volume dataVolumeNodeGroups2 = new Volume();
dataVolumeNodeGroups2.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups2 = new Volume();
rootVolumeNodeGroups2.withType("SAS")
    .withSize(480);
List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("master_node_default_group")
        .withNodeNum(2)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups2)
        .withDataVolume(dataVolumeNodeGroups2)
        .withDataVolumeCount(1)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("core_node_analysis_group")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups1)
        .withDataVolume(dataVolumeNodeGroups1)
        .withDataVolumeCount(1)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("task_node_analysis_group")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups)
        .withDataVolume(dataVolumeNodeGroups)
        .withDataVolumeCount(1)
        .withAutoScalingPolicy(autoScalingPolicyNodeGroups)
);
List<Tag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tag()
        .withKey("tag1")
        .withValue("111")
);
listbodyTags.add(
    new Tag()
        .withKey("tag2")
        .withValue("222")
);
ChargeInfo chargeInfobody = new ChargeInfo();
chargeInfobody.withChargeMode("postPaid");
body.withComponentConfigs(listbodyComponentConfigs);
body.withNodeGroups(listbodyNodeGroups);
body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
body.withTags(listbodyTags);
body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
body.withNodeRootPassword("your password");
body.withLoginMode("PASSWORD");
body.withManagerAdminPassword("your password");
body.withSafeMode("KERBEROS");
body.withAvailabilityZone("");
body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,Ranger,Tez");
body.withSubnetName("subnet");
```

```
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("ANALYSIS");
        body.withClusterName("mrs_DyJA_dm");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 创建一个流式集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3，一个Task节点组，节点数为0。每周一的12点至13点开启弹性伸缩。

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

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

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        Trigger triggerRules = new Trigger();
        triggerRules.withMetricName("StormSlotAvailablePercentage")
            .withMetricValue("100")
```

```
        .withComparisonOperator("LTOE")
        .withEvaluationPeriods(1);
    List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
    listAutoScalingPolicyRules.add(
        new Rule()
            .withName("default-expand-1")
            .withDescription("")
            .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
            .withCoolDownMinutes(5)
            .withScalingAdjustment(1)
            .withTrigger(triggerRules)
    );
    List<ResourcesPlan.EffectiveDaysEnum> listResourcesPlansEffectiveDays = new ArrayList<>();
    listResourcesPlansEffectiveDays.add(ResourcesPlan.EffectiveDaysEnum.fromValue("MONDAY"));
    List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
    listAutoScalingPolicyResourcesPlans.add(
        new ResourcesPlan()
            .withPeriodType("daily")
            .withStartTime("12:00")
            .withEndTime("13:00")
            .withMinCapacity(2)
            .withMaxCapacity(3)
            .withEffectiveDays(listResourcesPlansEffectiveDays)
    );
    AutoScalingPolicy autoScalingPolicyNodeGroups = new AutoScalingPolicy();
    autoScalingPolicyNodeGroups.withAutoScalingEnable(true)
        .withMinCapacity(0)
        .withMaxCapacity(1)
        .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
        .withRules(listAutoScalingPolicyRules);
    Volume dataVolumeNodeGroups = new Volume();
    dataVolumeNodeGroups.withType("SAS")
        .withSize(600);
    Volume rootVolumeNodeGroups = new Volume();
    rootVolumeNodeGroups.withType("SAS")
        .withSize(480);
    Volume dataVolumeNodeGroups1 = new Volume();
    dataVolumeNodeGroups1.withType("SAS")
        .withSize(600);
    Volume rootVolumeNodeGroups1 = new Volume();
    rootVolumeNodeGroups1.withType("SAS")
        .withSize(480);
    Volume dataVolumeNodeGroups2 = new Volume();
    dataVolumeNodeGroups2.withType("SAS")
        .withSize(600);
    Volume rootVolumeNodeGroups2 = new Volume();
    rootVolumeNodeGroups2.withType("SAS")
        .withSize(480);
    List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
    listbodyNodeGroups.add(
        new NodeGroupV2()
            .withGroupName("master_node_default_group")
            .withNodeNum(2)
            .withNodeSize("rc3.4xlarge.4.linux.bigdata")
            .withRootVolume(rootVolumeNodeGroups2)
            .withDataVolume(dataVolumeNodeGroups2)
            .withDataVolumeCount(1)
    );
    listbodyNodeGroups.add(
        new NodeGroupV2()
            .withGroupName("core_node_streaming_group")
            .withNodeNum(3)
            .withNodeSize("rc3.4xlarge.4.linux.bigdata")
            .withRootVolume(rootVolumeNodeGroups1)
            .withDataVolume(dataVolumeNodeGroups1)
            .withDataVolumeCount(1)
    );
    listbodyNodeGroups.add(
        new NodeGroupV2()
```

```
            .withGroupName("task_node_streaming_group")
            .withNodeNum(0)
            .withNodeSize("rc3.4xlarge.4.linux.bigdata")
            .withRootVolume(rootVolumeNodeGroups)
            .withDataVolume(dataVolumeNodeGroups)
            .withDataVolumeCount(1)
            .withAutoScalingPolicy(autoScalingPolicyNodeGroups)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("tag1")
                .withValue("111")
        );
        listbodyTags.add(
            new Tag()
                .withKey("tag2")
                .withValue("222")
        );
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");
        body.withComponents("Storm,Kafka,Flume,Ranger");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("STREAMING");
        body.withClusterName("mrs_Dokle_dm");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 创建一个混合集群，集群版本号为MRS 3.1.0。其中包含一个Master节点组，节点数为2；两个Core节点组，每个Core节点组的节点数均为3；两个Task节点组，一个Task节点组节点数为1，另一个节点数为0。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
```

```
import com.huaweicloud.sdk.mrs.v2.model.*;

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        Volume dataVolumeNodeGroups = new Volume();
        dataVolumeNodeGroups.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups = new Volume();
        rootVolumeNodeGroups.withType("SAS")
            .withSize(480);
        Volume dataVolumeNodeGroups1 = new Volume();
        dataVolumeNodeGroups1.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups1 = new Volume();
        rootVolumeNodeGroups1.withType("SAS")
            .withSize(480);
        Volume dataVolumeNodeGroups2 = new Volume();
        dataVolumeNodeGroups2.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups2 = new Volume();
        rootVolumeNodeGroups2.withType("SAS")
            .withSize(480);
        Volume dataVolumeNodeGroups3 = new Volume();
        dataVolumeNodeGroups3.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups3 = new Volume();
        rootVolumeNodeGroups3.withType("SAS")
            .withSize(480);
        Volume dataVolumeNodeGroups4 = new Volume();
        dataVolumeNodeGroups4.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups4 = new Volume();
        rootVolumeNodeGroups4.withType("SAS")
            .withSize(480);
        List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("master_node_default_group")
                .withNodeNum(2)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups4)
                .withDataVolume(dataVolumeNodeGroups4)
                .withDataVolumeCount(1)
        );
```

```
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("core_node_streaming_group")
                .withNodeNum(3)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups3)
                .withDataVolume(dataVolumeNodeGroups3)
                .withDataVolumeCount(1)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("core_node_analysis_group")
                .withNodeNum(3)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups2)
                .withDataVolume(dataVolumeNodeGroups2)
                .withDataVolumeCount(1)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("task_node_analysis_group")
                .withNodeNum(1)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups1)
                .withDataVolume(dataVolumeNodeGroups1)
                .withDataVolumeCount(1)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("task_node_streaming_group")
                .withNodeNum(0)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups)
                .withDataVolume(dataVolumeNodeGroups)
                .withDataVolumeCount(1)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("tag1")
                .withValue("111")
        );
        listbodyTags.add(
            new Tag()
                .withKey("tag2")
                .withValue("222")
        );
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");

body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,
Tez");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("MIXED");
        body.withClusterName("mrs_onmm_dm");
        body.withClusterVersion("MRS 3.1.0");
```

```
        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());
        }
    }
}
```

- 创建自定义管控合设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为3；两个Core节点组，一个节点数为3，另一个节点数为1。

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

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

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
        listNodeGroupsAssignedRoles.add("NodeManager");
        listNodeGroupsAssignedRoles.add("KerberosClient");
        listNodeGroupsAssignedRoles.add("SlapdClient");
        listNodeGroupsAssignedRoles.add("meta");
        listNodeGroupsAssignedRoles.add("FlinkResource");
        Volume dataVolumeNodeGroups = new Volume();
        dataVolumeNodeGroups.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups = new Volume();
        rootVolumeNodeGroups.withType("SAS")
```

```
    .withSize(480);
List<String> listNodeGroupsAssignedRoles1 = new ArrayList<>();
listNodeGroupsAssignedRoles1.add("DataNode");
listNodeGroupsAssignedRoles1.add("NodeManager");
listNodeGroupsAssignedRoles1.add("RegionServer");
listNodeGroupsAssignedRoles1.add("Flume:1");
listNodeGroupsAssignedRoles1.add("Broker");
listNodeGroupsAssignedRoles1.add("Supervisor");
listNodeGroupsAssignedRoles1.add("Logviewer");
listNodeGroupsAssignedRoles1.add("HBaseIndexer");
listNodeGroupsAssignedRoles1.add("KerberosClient");
listNodeGroupsAssignedRoles1.add("SlapdClient");
listNodeGroupsAssignedRoles1.add("meta");
listNodeGroupsAssignedRoles1.add("HSBroker:1,2");
listNodeGroupsAssignedRoles1.add("ThriftServer");
listNodeGroupsAssignedRoles1.add("ThriftServer1");
listNodeGroupsAssignedRoles1.add("RESTServer");
listNodeGroupsAssignedRoles1.add("FlinkResource");
Volume dataVolumeNodeGroups1 = new Volume();
dataVolumeNodeGroups1.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups1 = new Volume();
rootVolumeNodeGroups1.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles2 = new ArrayList<>();
listNodeGroupsAssignedRoles2.add("OMSServer:1,2");
listNodeGroupsAssignedRoles2.add("SlapdServer:1,2");
listNodeGroupsAssignedRoles2.add("KerberosServer:1,2");
listNodeGroupsAssignedRoles2.add("KerberosAdmin:1,2");
listNodeGroupsAssignedRoles2.add("quorumpeer:1,2,3");
listNodeGroupsAssignedRoles2.add("NameNode:2,3");
listNodeGroupsAssignedRoles2.add("Zkfc:2,3");
listNodeGroupsAssignedRoles2.add("JournalNode:1,2,3");
listNodeGroupsAssignedRoles2.add("ResourceManager:2,3");
listNodeGroupsAssignedRoles2.add("JobHistoryServer:2,3");
listNodeGroupsAssignedRoles2.add("DBServer:1,3");
listNodeGroupsAssignedRoles2.add("Hue:1,3");
listNodeGroupsAssignedRoles2.add("LoaderServer:1,3");
listNodeGroupsAssignedRoles2.add("MetaStore:1,2,3");
listNodeGroupsAssignedRoles2.add("WebHCat:1,2,3");
listNodeGroupsAssignedRoles2.add("HiveServer:1,2,3");
listNodeGroupsAssignedRoles2.add("HMaster:2,3");
listNodeGroupsAssignedRoles2.add("MonitorServer:1,2");
listNodeGroupsAssignedRoles2.add("Nimbus:1,2");
listNodeGroupsAssignedRoles2.add("UI:1,2");
listNodeGroupsAssignedRoles2.add("JDBCServer2x:1,2,3");
listNodeGroupsAssignedRoles2.add("JobHistory2x:2,3");
listNodeGroupsAssignedRoles2.add("SparkResource2x:1,2,3");
listNodeGroupsAssignedRoles2.add("oozie:2,3");
listNodeGroupsAssignedRoles2.add("LoadBalancer:2,3");
listNodeGroupsAssignedRoles2.add("TezUI:1,3");
listNodeGroupsAssignedRoles2.add("TimelineServer:3");
listNodeGroupsAssignedRoles2.add("RangerAdmin:1,2");
listNodeGroupsAssignedRoles2.add("UserSync:2");
listNodeGroupsAssignedRoles2.add("TagSync:2");
listNodeGroupsAssignedRoles2.add("KerberosClient");
listNodeGroupsAssignedRoles2.add("SlapdClient");
listNodeGroupsAssignedRoles2.add("meta");
listNodeGroupsAssignedRoles2.add("HSConsole:2,3");
listNodeGroupsAssignedRoles2.add("FlinkResource:1,2,3");
listNodeGroupsAssignedRoles2.add("DataNode:1,2,3");
listNodeGroupsAssignedRoles2.add("NodeManager:1,2,3");
listNodeGroupsAssignedRoles2.add("IndexServer2x:1,2");
listNodeGroupsAssignedRoles2.add("ThriftServer:1,2,3");
listNodeGroupsAssignedRoles2.add("RegionServer:1,2,3");
listNodeGroupsAssignedRoles2.add("ThriftServer1:1,2,3");
listNodeGroupsAssignedRoles2.add("RESTServer:1,2,3");
listNodeGroupsAssignedRoles2.add("Broker:1,2,3");
listNodeGroupsAssignedRoles2.add("Supervisor:1,2,3");
```

```
        listNodeGroupsAssignedRoles2.add("Logviewer:1,2,3");
        listNodeGroupsAssignedRoles2.add("Flume:1,2,3");
        listNodeGroupsAssignedRoles2.add("HSBroker:1,2,3");
        Volume dataVolumeNodeGroups2 = new Volume();
        dataVolumeNodeGroups2.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups2 = new Volume();
        rootVolumeNodeGroups2.withType("SAS")
            .withSize(480);
        List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("master_node_default_group")
                .withNodeNum(3)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups2)
                .withDataVolume(dataVolumeNodeGroups2)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles2)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("node_group_1")
                .withNodeNum(3)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups1)
                .withDataVolume(dataVolumeNodeGroups1)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles1)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("node_group_2")
                .withNodeNum(1)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups)
                .withDataVolume(dataVolumeNodeGroups)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("tag1")
                .withValue("111")
        );
        listbodyTags.add(
            new Tag()
                .withKey("tag2")
                .withValue("222")
        );
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withTemplateId("mgmt_control_combined_v2");
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");

body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,
Tez");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
```

```
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("CUSTOM");
        body.withClusterName("mrs_heshe_dm");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 创建自定义管控分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为5；一个Core节点组，节点数为3。

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

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

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
        listNodeGroupsAssignedRoles.add("DataNode");
        listNodeGroupsAssignedRoles.add("NodeManager");
        listNodeGroupsAssignedRoles.add("RegionServer");
        listNodeGroupsAssignedRoles.add("Flume:1");
        listNodeGroupsAssignedRoles.add("Broker");
```

```
listNodeGroupsAssignedRoles.add("Supervisor");
listNodeGroupsAssignedRoles.add("Logviewer");
listNodeGroupsAssignedRoles.add("HBaseIndexer");
listNodeGroupsAssignedRoles.add("KerberosClient");
listNodeGroupsAssignedRoles.add("SlapdClient");
listNodeGroupsAssignedRoles.add("meta");
listNodeGroupsAssignedRoles.add("HSBroker:1,2");
listNodeGroupsAssignedRoles.add("ThriftServer");
listNodeGroupsAssignedRoles.add("ThriftServer1");
listNodeGroupsAssignedRoles.add("RESTServer");
listNodeGroupsAssignedRoles.add("FlinkResource");
Volume dataVolumeNodeGroups = new Volume();
dataVolumeNodeGroups.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups = new Volume();
rootVolumeNodeGroups.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles1 = new ArrayList<>();
listNodeGroupsAssignedRoles1.add("OMSServer:1,2");
listNodeGroupsAssignedRoles1.add("SlapdServer:3,4");
listNodeGroupsAssignedRoles1.add("KerberosServer:3,4");
listNodeGroupsAssignedRoles1.add("KerberosAdmin:3,4");
listNodeGroupsAssignedRoles1.add("quorumpeer:3,4,5");
listNodeGroupsAssignedRoles1.add("NameNode:4,5");
listNodeGroupsAssignedRoles1.add("Zkfc:4,5");
listNodeGroupsAssignedRoles1.add("JournalNode:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("ResourceManager:4,5");
listNodeGroupsAssignedRoles1.add("JobHistoryServer:4,5");
listNodeGroupsAssignedRoles1.add("DBServer:3,5");
listNodeGroupsAssignedRoles1.add("Hue:1,2");
listNodeGroupsAssignedRoles1.add("LoaderServer:1,2");
listNodeGroupsAssignedRoles1.add("MetaStore:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("WebHCat:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("HiveServer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("HMaster:4,5");
listNodeGroupsAssignedRoles1.add("MonitorServer:1,2");
listNodeGroupsAssignedRoles1.add("Nimbus:1,2");
listNodeGroupsAssignedRoles1.add("UI:1,2");
listNodeGroupsAssignedRoles1.add("JDBCServer2x:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("JobHistory2x:4,5");
listNodeGroupsAssignedRoles1.add("SparkResource2x:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("oozie:1,2");
listNodeGroupsAssignedRoles1.add("LoadBalancer:1,2");
listNodeGroupsAssignedRoles1.add("TezUI:1,2");
listNodeGroupsAssignedRoles1.add("TimelineServer:5");
listNodeGroupsAssignedRoles1.add("RangerAdmin:1,2");
listNodeGroupsAssignedRoles1.add("KerberosClient");
listNodeGroupsAssignedRoles1.add("SlapdClient");
listNodeGroupsAssignedRoles1.add("meta");
listNodeGroupsAssignedRoles1.add("HSConsole:1,2");
listNodeGroupsAssignedRoles1.add("FlinkResource:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("DataNode:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("NodeManager:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("IndexServer2x:1,2");
listNodeGroupsAssignedRoles1.add("ThriftServer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("RegionServer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("ThriftServer1:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("RESTServer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("Broker:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("Supervisor:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("Logviewer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("Flume:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("HBaseIndexer:1,2,3,4,5");
listNodeGroupsAssignedRoles1.add("TagSync:1");
listNodeGroupsAssignedRoles1.add("UserSync:1");
Volume dataVolumeNodeGroups1 = new Volume();
dataVolumeNodeGroups1.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups1 = new Volume();
```

```
        rootVolumeNodeGroups1.withType("SAS")
            .withSize(480);
        List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("master_node_default_group")
                .withNodeNum(5)
                .withNodeSize("rc3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups1)
                .withDataVolume(dataVolumeNodeGroups1)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles1)
        );
        listbodyNodeGroups.add(
            new NodeGroupV2()
                .withGroupName("node_group_1")
                .withNodeNum(3)
                .withNodeSize("rc3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups)
                .withDataVolume(dataVolumeNodeGroups)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("aaa")
                .withValue("111")
        );
        listbodyTags.add(
            new Tag()
                .withKey("bbb")
                .withValue("222")
        );
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withTemplateId("mgmt_control_separated_v2");
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");

body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,
Tez");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("CUSTOM");
        body.withClusterName("mrs_jdRU_dm01");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 创建自定义数据分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为9；四个Core节点组，每个Core节点组的节点数均为3。

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

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

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV2 body = new CreateClusterReqV2();
        List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
        listNodeGroupsAssignedRoles.add("Broker");
        listNodeGroupsAssignedRoles.add("Supervisor");
        listNodeGroupsAssignedRoles.add("Logviewer");
        listNodeGroupsAssignedRoles.add("KerberosClient");
        listNodeGroupsAssignedRoles.add("SlapdClient");
        listNodeGroupsAssignedRoles.add("meta");
        Volume dataVolumeNodeGroups = new Volume();
        dataVolumeNodeGroups.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups = new Volume();
        rootVolumeNodeGroups.withType("SAS")
            .withSize(480);
        List<String> listNodeGroupsAssignedRoles1 = new ArrayList<>();
        listNodeGroupsAssignedRoles1.add("KerberosClient");
        listNodeGroupsAssignedRoles1.add("SlapdClient");
        listNodeGroupsAssignedRoles1.add("meta");
        Volume dataVolumeNodeGroups1 = new Volume();
        dataVolumeNodeGroups1.withType("SAS")
            .withSize(600);
        Volume rootVolumeNodeGroups1 = new Volume();
        rootVolumeNodeGroups1.withType("SAS")
            .withSize(480);
```

```
List<String> listNodeGroupsAssignedRoles2 = new ArrayList<>();
listNodeGroupsAssignedRoles2.add("HBaseIndexer");
listNodeGroupsAssignedRoles2.add("SolrServer[3]");
listNodeGroupsAssignedRoles2.add("EsNode[2]");
listNodeGroupsAssignedRoles2.add("KerberosClient");
listNodeGroupsAssignedRoles2.add("SlapdClient");
listNodeGroupsAssignedRoles2.add("meta");
listNodeGroupsAssignedRoles2.add("SolrServerAdmin:1,2");
Volume dataVolumeNodeGroups2 = new Volume();
dataVolumeNodeGroups2.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups2 = new Volume();
rootVolumeNodeGroups2.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles3 = new ArrayList<>();
listNodeGroupsAssignedRoles3.add("DataNode");
listNodeGroupsAssignedRoles3.add("NodeManager");
listNodeGroupsAssignedRoles3.add("RegionServer");
listNodeGroupsAssignedRoles3.add("Flume:1");
listNodeGroupsAssignedRoles3.add("GraphServer");
listNodeGroupsAssignedRoles3.add("KerberosClient");
listNodeGroupsAssignedRoles3.add("SlapdClient");
listNodeGroupsAssignedRoles3.add("meta");
listNodeGroupsAssignedRoles3.add("HSBroker:1,2");
Volume dataVolumeNodeGroups3 = new Volume();
dataVolumeNodeGroups3.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups3 = new Volume();
rootVolumeNodeGroups3.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles4 = new ArrayList<>();
listNodeGroupsAssignedRoles4.add("OMSServer:1,2");
listNodeGroupsAssignedRoles4.add("SlapdServer:5,6");
listNodeGroupsAssignedRoles4.add("KerberosServer:5,6");
listNodeGroupsAssignedRoles4.add("KerberosAdmin:5,6");
listNodeGroupsAssignedRoles4.add("quorumpeer:5,6,7,8,9");
listNodeGroupsAssignedRoles4.add("NameNode:3,4");
listNodeGroupsAssignedRoles4.add("Zkfc:3,4");
listNodeGroupsAssignedRoles4.add("JournalNode:5,6,7");
listNodeGroupsAssignedRoles4.add("ResourceManager:8,9");
listNodeGroupsAssignedRoles4.add("JobHistoryServer:8");
listNodeGroupsAssignedRoles4.add("DBServer:8,9");
listNodeGroupsAssignedRoles4.add("Hue:8,9");
listNodeGroupsAssignedRoles4.add("FlinkResource:3,4");
listNodeGroupsAssignedRoles4.add("LoaderServer:3,5");
listNodeGroupsAssignedRoles4.add("MetaStore:8,9");
listNodeGroupsAssignedRoles4.add("WebHCat:5");
listNodeGroupsAssignedRoles4.add("HiveServer:8,9");
listNodeGroupsAssignedRoles4.add("HMaster:8,9");
listNodeGroupsAssignedRoles4.add("FTP-Server:3,4");
listNodeGroupsAssignedRoles4.add("MonitorServer:3,4");
listNodeGroupsAssignedRoles4.add("Nimbus:8,9");
listNodeGroupsAssignedRoles4.add("UI:8,9");
listNodeGroupsAssignedRoles4.add("JDBCServer2x:8,9");
listNodeGroupsAssignedRoles4.add("JobHistory2x:8,9");
listNodeGroupsAssignedRoles4.add("SparkResource2x:5,6,7");
listNodeGroupsAssignedRoles4.add("oozie:4,5");
listNodeGroupsAssignedRoles4.add("EsMaster:7,8,9");
listNodeGroupsAssignedRoles4.add("LoadBalancer:8,9");
listNodeGroupsAssignedRoles4.add("TezUI:5,6");
listNodeGroupsAssignedRoles4.add("TimelineServer:5");
listNodeGroupsAssignedRoles4.add("RangerAdmin:4,5");
listNodeGroupsAssignedRoles4.add("UserSync:5");
listNodeGroupsAssignedRoles4.add("TagSync:5");
listNodeGroupsAssignedRoles4.add("KerberosClient");
listNodeGroupsAssignedRoles4.add("SlapdClient");
listNodeGroupsAssignedRoles4.add("meta");
listNodeGroupsAssignedRoles4.add("HSBroker:5");
listNodeGroupsAssignedRoles4.add("HSConsole:3,4");
```

```
listNodeGroupsAssignedRoles4.add("FlinkResource:3,4");
Volume dataVolumeNodeGroups4 = new Volume();
dataVolumeNodeGroups4.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups4 = new Volume();
rootVolumeNodeGroups4.withType("SAS")
    .withSize(480);
List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("master_node_default_group")
        .withNodeNum(9)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups4)
        .withDataVolume(dataVolumeNodeGroups4)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles4)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("node_group_1")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups3)
        .withDataVolume(dataVolumeNodeGroups3)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles3)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("node_group_2")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups2)
        .withDataVolume(dataVolumeNodeGroups2)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles2)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("node_group_3")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups1)
        .withDataVolume(dataVolumeNodeGroups1)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles1)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("node_group_4")
        .withNodeNum(3)
        .withNodeSize("rc3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups)
        .withDataVolume(dataVolumeNodeGroups)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles)
);
List<Tag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tag()
        .withKey("aaa")
        .withValue("111")
);
listbodyTags.add(
    new Tag()
        .withKey("bbb")
        .withValue("222")
);
```

```
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(CreateClusterReqV2.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withTemplateId("mgmt_control_data_separated_v2");
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");

body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,
Tez");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("CUSTOM");
        body.withClusterName("mrs_jdRU_dm02");
        body.withClusterVersion("MRS 3.1.0");
        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

- 创建一个分析集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3；一个Task节点组，节点数为3。每周一的12点至13点开启弹性伸缩。Hive组件的初始配置hive.union.data.type.incompatible.enable修改为true，dfs.replication修改为4。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

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

try:
    request = CreateClusterRequest()
    listConfigsComponentConfigs = [
        Config(
            key="hive.union.data.type.incompatible.enable",
            value="true",
            config_file_name="hive-site.xml"
        ),
        Config(
            key="dfs.replication",
            value="4",
            config_file_name="hdfs-site.xml"
        )
    ]
    listComponentConfigsbody = [
        ComponentConfig(
            component_name="Hive",
            configs=listConfigsComponentConfigs
        )
    ]
    listNodesExecScripts = [
        "master_node_default_group",
        "core_node_analysis_group",
        "task_node_analysis_group"
    ]
    listExecScriptsAutoScalingPolicy = [
        ScaleScript(
            name="test",
            uri="s3a://obs-mrstest/bootstrap/basic_success.sh",
            parameters="",
            nodes=listNodesExecScripts,
            active_master=False,
            fail_action="continue",
            action_stage="before_scale_out"
        )
    ]
    triggerRules = Trigger(
        metric_name="YARNAppRunning",
        metric_value="100",
        comparison_operator="GTOE",
        evaluation_periods=1
    )
    listRulesAutoScalingPolicy = [
        Rule(
            name="default-expand-1",
            description="",
            adjustment_type="scale_out",
            cool_down_minutes=5,
            scaling_adjustment=1,
            trigger=triggerRules
        )
    ]
    listEffectiveDaysResourcesPlans = [
        "MONDAY"
    ]
    listResourcesPlansAutoScalingPolicy = [
        ResourcesPlan(
            period_type="daily",
            start_time="12:00",
            end_time="13:00",
            min_capacity=2,
            max_capacity=3,
            effective_days=listEffectiveDaysResourcesPlans
        )
```

```
]
autoScalingPolicyNodeGroups = AutoScalingPolicy(
    auto_scaling_enable=True,
    min_capacity=0,
    max_capacity=1,
    resources_plans=listResourcesPlansAutoScalingPolicy,
    rules=listRulesAutoScalingPolicy,
    exec_scripts=listExecScriptsAutoScalingPolicy
)
dataVolumeNodeGroups = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups = Volume(
    type="SAS",
    size=480
)
dataVolumeNodeGroups1 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups1 = Volume(
    type="SAS",
    size=480
)
dataVolumeNodeGroups2 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups2 = Volume(
    type="SAS",
    size=480
)
listNodeGroupsbody = [
    NodeGroupV2(
        group_name="master_node_default_group",
        node_num=2,
        node_size="rc3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups2,
        data_volume=dataVolumeNodeGroups2,
        data_volume_count=1
    ),
    NodeGroupV2(
        group_name="core_node_analysis_group",
        node_num=3,
        node_size="rc3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups1,
        data_volume=dataVolumeNodeGroups1,
        data_volume_count=1
    ),
    NodeGroupV2(
        group_name="task_node_analysis_group",
        node_num=3,
        node_size="rc3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups,
        data_volume=dataVolumeNodeGroups,
        data_volume_count=1,
        auto_scaling_policy=autoScalingPolicyNodeGroups
    )
]
listTagsbody = [
    Tag(
        key="tag1",
        value="111"
    ),
    Tag(
        key="tag2",
        value="222"
    )
```

```
        ]
        chargeInfobody = ChargeInfo(
            charge_mode="postPaid"
        )
        request.body = CreateClusterReqV2(
            component_configs=listComponentConfigsbody,
            node_groups=listNodeGroupsbody,
            log_collection=1,
            tags=listTagsbody,
            mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
            node_root_password="your password",
            login_mode="PASSWORD",
            manager_admin_password="your password",
            safe_mode="KERBEROS",
            availability_zone="",
            components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,Ranger,Tez",
            subnet_name="subnet",
            subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
            vpc_name="vpc-37cd",
            region="",
            charge_info=chargeInfobody,
            cluster_type="ANALYSIS",
            cluster_name="mrs_DyJA_dm",
            cluster_version="MRS 3.1.0"
        )
        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)
```

- 创建一个流式集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3，一个Task节点组，节点数为0。每周一的12点至13点开启弹性伸缩。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        triggerRules = Trigger(
            metric_name="StormSlotAvailablePercentage",
            metric_value="100",
            comparison_operator="LTOE",
            evaluation_periods=1
```

```
)
listRulesAutoScalingPolicy = [
    Rule(
        name="default-expand-1",
        description="",
        adjustment_type="scale_out",
        cool_down_minutes=5,
        scaling_adjustment=1,
        trigger=triggerRules
    )
]
listEffectiveDaysResourcesPlans = [
    "MONDAY"
]
listResourcesPlansAutoScalingPolicy = [
    ResourcesPlan(
        period_type="daily",
        start_time="12:00",
        end_time="13:00",
        min_capacity=2,
        max_capacity=3,
        effective_days=listEffectiveDaysResourcesPlans
    )
]
autoScalingPolicyNodeGroups = AutoScalingPolicy(
    auto_scaling_enable=True,
    min_capacity=0,
    max_capacity=1,
    resources_plans=listResourcesPlansAutoScalingPolicy,
    rules=listRulesAutoScalingPolicy
)
dataVolumeNodeGroups = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups = Volume(
    type="SAS",
    size=480
)
dataVolumeNodeGroups1 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups1 = Volume(
    type="SAS",
    size=480
)
dataVolumeNodeGroups2 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups2 = Volume(
    type="SAS",
    size=480
)
listNodeGroupsbody = [
    NodeGroupV2(
        group_name="master_node_default_group",
        node_num=2,
        node_size="rc3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups2,
        data_volume=dataVolumeNodeGroups2,
        data_volume_count=1
    ),
    NodeGroupV2(
        group_name="core_node_streaming_group",
        node_num=3,
        node_size="rc3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups1,
```

```
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1
        ),
        NodeGroupV2(
            group_name="task_node_streaming_group",
            node_num=0,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1,
            auto_scaling_policy=autoScalingPolicyNodeGroups
        )
    ]
    listTagsbody = [
        Tag(
            key="tag1",
            value="111"
        ),
        Tag(
            key="tag2",
            value="222"
        )
    ]
    chargeInfobody = ChargeInfo(
        charge_mode="postPaid"
    )
    request.body = CreateClusterReqV2(
        node_groups=listNodeGroupsbody,
        log_collection=1,
        tags=listTagsbody,
        mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
        node_root_password="your password",
        login_mode="PASSWORD",
        manager_admin_password="your password",
        safe_mode="KERBEROS",
        availability_zone="",
        components="Storm,Kafka,Flume,Ranger",
        subnet_name="subnet",
        subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
        vpc_name="vpc-37cd",
        region="",
        charge_info=chargeInfobody,
        cluster_type="STREAMING",
        cluster_name="mrs_Dokle_dm",
        cluster_version="MRS 3.1.0"
    )
    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)
```

- 创建一个混合集群，集群版本号为MRS 3.1.0。其中包含一个Master节点组，节点数为2；两个Core节点组，每个Core节点组的节点数均为3；两个Task节点组，一个Task节点组节点数为1，另一个节点数为0。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        dataVolumeNodeGroups = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups = Volume(
            type="SAS",
            size=480
        )
        dataVolumeNodeGroups1 = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups1 = Volume(
            type="SAS",
            size=480
        )
        dataVolumeNodeGroups2 = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups2 = Volume(
            type="SAS",
            size=480
        )
        dataVolumeNodeGroups3 = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups3 = Volume(
            type="SAS",
            size=480
        )
        dataVolumeNodeGroups4 = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups4 = Volume(
            type="SAS",
            size=480
        )
        listNodeGroupsbody = [
            NodeGroupV2(
                group_name="master_node_default_group",
                node_num=2,
                node_size="Sit3.4xlarge.4.linux.bigdata",
                root_volume=rootVolumeNodeGroups4,
                data_volume=dataVolumeNodeGroups4,
                data_volume_count=1
            ),
            NodeGroupV2(
                group_name="core_node_streaming_group",
                node_num=3,
                node_size="Sit3.4xlarge.4.linux.bigdata",
```

```
            root_volume=rootVolumeNodeGroups3,
            data_volume=dataVolumeNodeGroups3,
            data_volume_count=1
        ),
        NodeGroupV2(
            group_name="core_node_analysis_group",
            node_num=3,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups2,
            data_volume=dataVolumeNodeGroups2,
            data_volume_count=1
        ),
        NodeGroupV2(
            group_name="task_node_analysis_group",
            node_num=1,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups1,
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1
        ),
        NodeGroupV2(
            group_name="task_node_streaming_group",
            node_num=0,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1
        )
    ]
    listTagsbody = [
        Tag(
            key="tag1",
            value="111"
        ),
        Tag(
            key="tag2",
            value="222"
        )
    ]
    chargeInfobody = ChargeInfo(
        charge_mode="postPaid"
    )
    request.body = CreateClusterReqV2(
        node_groups=listNodeGroupsbody,
        log_collection=1,
        tags=listTagsbody,
        mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
        node_root_password="your password",
        login_mode="PASSWORD",
        manager_admin_password="your password",
        safe_mode="KERBEROS",
        availability_zone="",
components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        subnet_name="subnet",
        subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
        vpc_name="vpc-37cd",
        region="",
        charge_info=chargeInfobody,
        cluster_type="MIXED",
        cluster_name="mrs_onmm_dm",
        cluster_version="MRS 3.1.0"
    )
    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)
```

- 创建自定义管控合设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为3；两个Core节点组，一个节点数为3，另一个节点数为1。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listAssignedRolesNodeGroups = [
            "NodeManager",
            "KerberosClient",
            "SlapdClient",
            "meta",
            "FlinkResource"
        ]
        dataVolumeNodeGroups = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups = Volume(
            type="SAS",
            size=480
        )
        listAssignedRolesNodeGroups1 = [
            "DataNode",
            "NodeManager",
            "RegionServer",
            "Flume:1",
            "Broker",
            "Supervisor",
            "Logviewer",
            "HBaseIndexer",
            "KerberosClient",
            "SlapdClient",
            "meta",
            "HSBroker:1,2",
            "ThriftServer",
            "ThriftServer1",
            "RESTServer",
            "FlinkResource"
        ]
        dataVolumeNodeGroups1 = Volume(
            type="SAS",
            size=600
```

```
)
rootVolumeNodeGroups1 = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups2 = [
    "OMSServer:1,2",
    "SlapdServer:1,2",
    "KerberosServer:1,2",
    "KerberosAdmin:1,2",
    "quorumpeer:1,2,3",
    "NameNode:2,3",
    "Zkfc:2,3",
    "JournalNode:1,2,3",
    "ResourceManager:2,3",
    "JobHistoryServer:2,3",
    "DBServer:1,3",
    "Hue:1,3",
    "LoaderServer:1,3",
    "MetaStore:1,2,3",
    "WebHCat:1,2,3",
    "HiveServer:1,2,3",
    "HMaster:2,3",
    "MonitorServer:1,2",
    "Nimbus:1,2",
    "UI:1,2",
    "JDBCServer2x:1,2,3",
    "JobHistory2x:2,3",
    "SparkResource2x:1,2,3",
    "oozie:2,3",
    "LoadBalancer:2,3",
    "TezUI:1,3",
    "TimelineServer:3",
    "RangerAdmin:1,2",
    "UserSync:2",
    "TagSync:2",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSConsole:2,3",
    "FlinkResource:1,2,3",
    "DataNode:1,2,3",
    "NodeManager:1,2,3",
    "IndexServer2x:1,2",
    "ThriftServer:1,2,3",
    "RegionServer:1,2,3",
    "ThriftServer1:1,2,3",
    "RESTServer:1,2,3",
    "Broker:1,2,3",
    "Supervisor:1,2,3",
    "Logviewer:1,2,3",
    "Flume:1,2,3",
    "HSBroker:1,2,3"
]
dataVolumeNodeGroups2 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups2 = Volume(
    type="SAS",
    size=480
)
listNodeGroupsbody = [
    NodeGroupV2(
        group_name="master_node_default_group",
        node_num=3,
        node_size="Sit3.4xlarge.4.linux.bigdata",
        root_volume=rootVolumeNodeGroups2,
        data_volume=dataVolumeNodeGroups2,
```

```
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups2
        ),
        NodeGroupV2(
            group_name="node_group_1",
            node_num=3,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups1,
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups1
        ),
        NodeGroupV2(
            group_name="node_group_2",
            node_num=1,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups
        )
    ]
    listTagsbody = [
        Tag(
            key="tag1",
            value="111"
        ),
        Tag(
            key="tag2",
            value="222"
        )
    ]
    chargeInfobody = ChargeInfo(
        charge_mode="postPaid"
    )
    request.body = CreateClusterReqV2(
        node_groups=listNodeGroupsbody,
        log_collection=1,
        tags=listTagsbody,
        template_id="mgmt_control_combined_v2",
        mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
        node_root_password="your password",
        login_mode="PASSWORD",
        manager_admin_password="your password",
        safe_mode="KERBEROS",
        availability_zone="",
components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        subnet_name="subnet",
        subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
        vpc_name="vpc-37cd",
        region="",
        charge_info=chargeInfobody,
        cluster_type="CUSTOM",
        cluster_name="mrs_heshe_dm",
        cluster_version="MRS 3.1.0"
    )
    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)
```

- 创建自定义管控分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为5；一个Core节点组，节点数为3。

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

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listAssignedRolesNodeGroups = [
            "DataNode",
            "NodeManager",
            "RegionServer",
            "Flume:1",
            "Broker",
            "Supervisor",
            "Logviewer",
            "HBaseIndexer",
            "KerberosClient",
            "SlapdClient",
            "meta",
            "HSBroker:1,2",
            "ThriftServer",
            "ThriftServer1",
            "RESTServer",
            "FlinkResource"
        ]
        dataVolumeNodeGroups = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups = Volume(
            type="SAS",
            size=480
        )
        listAssignedRolesNodeGroups1 = [
            "OMSServer:1,2",
            "SlapdServer:3,4",
            "KerberosServer:3,4",
            "KerberosAdmin:3,4",
            "quorumpeer:3,4,5",
            "NameNode:4,5",
            "Zkfc:4,5",
            "JournalNode:1,2,3,4,5",
            "ResourceManager:4,5",
            "JobHistoryServer:4,5",
            "DBServer:3,5",
            "Hue:1,2",
            "LoaderServer:1,2",
            "MetaStore:1,2,3,4,5",
            "WebHCat:1,2,3,4,5",
            "HiveServer:1,2,3,4,5",
```

```
        "HMaster:4,5",
        "MonitorServer:1,2",
        "Nimbus:1,2",
        "UI:1,2",
        "JDBCServer2x:1,2,3,4,5",
        "JobHistory2x:4,5",
        "SparkResource2x:1,2,3,4,5",
        "oozie:1,2",
        "LoadBalancer:1,2",
        "TezUI:1,2",
        "TimelineServer:5",
        "RangerAdmin:1,2",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "HSConsole:1,2",
        "FlinkResource:1,2,3,4,5",
        "DataNode:1,2,3,4,5",
        "NodeManager:1,2,3,4,5",
        "IndexServer2x:1,2",
        "ThriftServer:1,2,3,4,5",
        "RegionServer:1,2,3,4,5",
        "ThriftServer1:1,2,3,4,5",
        "RESTServer:1,2,3,4,5",
        "Broker:1,2,3,4,5",
        "Supervisor:1,2,3,4,5",
        "Logviewer:1,2,3,4,5",
        "Flume:1,2,3,4,5",
        "HBaseIndexer:1,2,3,4,5",
        "TagSync:1",
        "UserSync:1"
    ]
    dataVolumeNodeGroups1 = Volume(
        type="SAS",
        size=600
    )
    rootVolumeNodeGroups1 = Volume(
        type="SAS",
        size=480
    )
    listNodeGroupsbody = [
        NodeGroupV2(
            group_name="master_node_default_group",
            node_num=5,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups1,
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups1
        ),
        NodeGroupV2(
            group_name="node_group_1",
            node_num=3,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups
        )
    ]
    listTagsbody = [
        Tag(
            key="aaa",
            value="111"
        ),
        Tag(
            key="bbb",
            value="222"
        )
```

```
        ]
        chargeInfobody = ChargeInfo(
            charge_mode="postPaid"
        )
        request.body = CreateClusterReqV2(
            node_groups=listNodeGroupsbody,
            log_collection=1,
            tags=listTagsbody,
            template_id="mgmt_control_separated_v2",
            mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
            node_root_password="your password",
            login_mode="PASSWORD",
            manager_admin_password="your password",
            safe_mode="KERBEROS",
            availability_zone="",

components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
            subnet_name="subnet",
            subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
            vpc_name="vpc-37cd",
            region="",
            charge_info=chargeInfobody,
            cluster_type="CUSTOM",
            cluster_name="mrs_jdRU_dm01",
            cluster_version="MRS 3.1.0"
        )
        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)
```

- 创建自定义数据分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为9；四个Core节点组，每个Core节点组的节点数均为3。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listAssignedRolesNodeGroups = [
            "Broker",
            "Supervisor",
            "Logviewer",
            "KerberosClient",
```

```
    "SlapdClient",
    "meta"
]
dataVolumeNodeGroups = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups1 = [
    "KerberosClient",
    "SlapdClient",
    "meta"
]
dataVolumeNodeGroups1 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups1 = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups2 = [
    "HBaseIndexer",
    "SolrServer[3]",
    "EsNode[2]",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "SolrServerAdmin:1,2"
]
dataVolumeNodeGroups2 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups2 = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups3 = [
    "DataNode",
    "NodeManager",
    "RegionServer",
    "Flume:1",
    "GraphServer",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSBroker:1,2"
]
dataVolumeNodeGroups3 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups3 = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups4 = [
    "OMSServer:1,2",
    "SlapdServer:5,6",
    "KerberosServer:5,6",
    "KerberosAdmin:5,6",
    "quorumpeer:5,6,7,8,9",
    "NameNode:3,4",
    "Zkfc:3,4",
    "JournalNode:5,6,7",
```

```
        "ResourceManager:8,9",
        "JobHistoryServer:8",
        "DBServer:8,9",
        "Hue:8,9",
        "FlinkResource:3,4",
        "LoaderServer:3,5",
        "MetaStore:8,9",
        "WebHCat:5",
        "HiveServer:8,9",
        "HMaster:8,9",
        "FTP-Server:3,4",
        "MonitorServer:3,4",
        "Nimbus:8,9",
        "UI:8,9",
        "JDBCServer2x:8,9",
        "JobHistory2x:8,9",
        "SparkResource2x:5,6,7",
        "oozie:4,5",
        "EsMaster:7,8,9",
        "LoadBalancer:8,9",
        "TezUI:5,6",
        "TimelineServer:5",
        "RangerAdmin:4,5",
        "UserSync:5",
        "TagSync:5",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "HSBroker:5",
        "HSConsole:3,4",
        "FlinkResource:3,4"
    ]
    dataVolumeNodeGroups4 = Volume(
        type="SAS",
        size=600
    )
    rootVolumeNodeGroups4 = Volume(
        type="SAS",
        size=480
    )
    listNodeGroupsbody = [
        NodeGroupV2(
            group_name="master_node_default_group",
            node_num=9,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups4,
            data_volume=dataVolumeNodeGroups4,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups4
        ),
        NodeGroupV2(
            group_name="node_group_1",
            node_num=3,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups3,
            data_volume=dataVolumeNodeGroups3,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups3
        ),
        NodeGroupV2(
            group_name="node_group_2",
            node_num=3,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups2,
            data_volume=dataVolumeNodeGroups2,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups2
        ),
        NodeGroupV2(
```

```
            group_name="node_group_3",
            node_num=3,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups1,
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups1
        ),
        NodeGroupV2(
            group_name="node_group_4",
            node_num=3,
            node_size="rc3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups
        )
    ]
    listTagsbody = [
        Tag(
            key="aaa",
            value="111"
        ),
        Tag(
            key="bbb",
            value="222"
        )
    ]
    chargeInfobody = ChargeInfo(
        charge_mode="postPaid"
    )
    request.body = CreateClusterReqV2(
        node_groups=listNodeGroupsbody,
        log_collection=1,
        tags=listTagsbody,
        template_id="mgmt_control_data_separated_v2",
        mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
        node_root_password="your password",
        login_mode="PASSWORD",
        manager_admin_password="your password",
        safe_mode="KERBEROS",
        availability_zone="",

components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        subnet_name="subnet",
        subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
        vpc_name="vpc-37cd",
        region="",
        charge_info=chargeInfobody,
        cluster_type="CUSTOM",
        cluster_name="mrs_jdRU_dm02",
        cluster_version="MRS 3.1.0"
    )
    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

- 创建一个分析集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3；一个Task节点组，节点数为3。每周一的12点至13点开启弹性伸缩。Hive组件的初始配置hive.union.data.type.incompatible.enable修改为true，dfs.replication修改为4。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listConfigsComponentConfigs = []model.Config{
        {
            Key: "hive.union.data.type.incompatible.enable",
            Value: "true",
            ConfigFileName: "hive-site.xml",
        },
        {
            Key: "dfs.replication",
            Value: "4",
            ConfigFileName: "hdfs-site.xml",
        },
    }
    var listComponentConfigsbody = []model.ComponentConfig{
        {
            ComponentName: "Hive",
            Configs: &listConfigsComponentConfigs,
        },
    }
    var listNodesExecScripts = []string{
        "master_node_default_group",
        "core_node_analysis_group",
        "task_node_analysis_group",
    }
    parametersExecScripts:= ""
    activeMasterExecScripts:= false
    var listExecScriptsAutoScalingPolicy = []model.ScaleScript{
        {
            Name: "test",
            Uri: "s3a://obs-mrstest/bootstrap/basic_success.sh",
            Parameters: &parametersExecScripts,
            Nodes: listNodesExecScripts,
            ActiveMaster: &activeMasterExecScripts,
            FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
            ActionStage: model.GetScaleScriptActionStageEnum().BEFORE_SCALE_OUT,
        },
```

```
}
comparisonOperatorTrigger:= "GTOE"
triggerRules := &model.Trigger{
    MetricName: "YARNAppRunning",
    MetricValue: "100",
    ComparisonOperator: &comparisonOperatorTrigger,
    EvaluationPeriods: int32(1),
}
descriptionRules:= ""
var listRulesAutoScalingPolicy = []model.Rule{
    {
        Name: "default-expand-1",
        Description: &descriptionRules,
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
        CoolDownMinutes: int32(5),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules,
    },
}
var listEffectiveDaysResourcesPlans = []model.ResourcesPlanEffectiveDays{
    model.GetResourcesPlanEffectiveDaysEnum().MONDAY,
}
var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
    {
        PeriodType: "daily",
        StartTime: "12:00",
        EndTime: "13:00",
        MinCapacity: int32(2),
        MaxCapacity: int32(3),
        EffectiveDays: &listEffectiveDaysResourcesPlans,
    },
}
autoScalingPolicyNodeGroups := &model.AutoScalingPolicy{
    AutoScalingEnable: true,
    MinCapacity: int32(0),
    MaxCapacity: int32(1),
    ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
    Rules: &listRulesAutoScalingPolicy,
    ExecScripts: &listExecScriptsAutoScalingPolicy,
}
dataVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeCountNodeGroups2:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
```

```
        GroupName: "master_node_default_group",
        NodeNum: int32(2),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups2,
        DataVolume: dataVolumeNodeGroups2,
        DataVolumeCount: &dataVolumeCountNodeGroups,
    },
    {
        GroupName: "core_node_analysis_group",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups1,
        DataVolume: dataVolumeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
    },
    {
        GroupName: "task_node_analysis_group",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups,
        DataVolume: dataVolumeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups2,
        AutoScalingPolicy: autoScalingPolicyNodeGroups,
    },
}
var listTagsbody = []model.Tag{
    {
        Key: "tag1",
        Value: "111",
    },
    {
        Key: "tag2",
        Value: "222",
    },
}
chargeInfobody := &model.ChargeInfo{
    ChargeMode: "postPaid",
}
logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
nodeRootPasswordCreateClusterReqV2:= "your password"
subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
request.Body = &model.CreateClusterReqV2{
    ComponentConfigs: &listComponentConfigsbody,
    NodeGroups: listNodeGroupsbody,
    LogCollection: &logCollectionCreateClusterReqV2,
    Tags: &listTagsbody,
    MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
    NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
    LoginMode: "PASSWORD",
    ManagerAdminPassword: "your password",
    SafeMode: "KERBEROS",
    AvailabilityZone: "",
    Components: "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,Ranger,Tez",
    SubnetName: "subnet",
    SubnetId: &subnetIdCreateClusterReqV2,
    VpcName: "vpc-37cd",
    Region: "",
    ChargeInfo: chargeInfobody,
    ClusterType: "ANALYSIS",
    ClusterName: "mrs_DyJA_dm",
    ClusterVersion: "MRS 3.1.0",
}
response, err := client.CreateCluster(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
```

```
    }
}
```

- 创建一个流式集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为2；一个Core节点组，节点数为3，一个Task节点组，节点数为0。每周一的12点至13点开启弹性伸缩。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    comparisonOperatorTrigger:= "LTOE"
    triggerRules := &model.Trigger{
        MetricName: "StormSlotAvailablePercentage",
        MetricValue: "100",
        ComparisonOperator: &comparisonOperatorTrigger,
        EvaluationPeriods: int32(1),
    }
    descriptionRules:= ""
    var listRulesAutoScalingPolicy = []model.Rule{
        {
            Name: "default-expand-1",
            Description: &descriptionRules,
            AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
            CoolDownMinutes: int32(5),
            ScalingAdjustment: int32(1),
            Trigger: triggerRules,
        },
    }
    var listEffectiveDaysResourcesPlans = []model.ResourcesPlanEffectiveDays{
        model.GetResourcesPlanEffectiveDaysEnum().MONDAY,
    }
    var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
        {
            PeriodType: "daily",
            StartTime: "12:00",
            EndTime: "13:00",
            MinCapacity: int32(2),
            MaxCapacity: int32(3),
            EffectiveDays: &listEffectiveDaysResourcesPlans,
```

```
    },
}
autoScalingPolicyNodeGroups := &model.AutoScalingPolicy{
    AutoScalingEnable: true,
    MinCapacity: int32(0),
    MaxCapacity: int32(1),
    ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
    Rules: &listRulesAutoScalingPolicy,
}
dataVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeCountNodeGroups2:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(2),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups2,
        DataVolume: dataVolumeNodeGroups2,
        DataVolumeCount: &dataVolumeCountNodeGroups,
    },
    {
        GroupName: "core_node_streaming_group",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups1,
        DataVolume: dataVolumeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
    },
    {
        GroupName: "task_node_streaming_group",
        NodeNum: int32(0),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups,
        DataVolume: dataVolumeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups2,
        AutoScalingPolicy: autoScalingPolicyNodeGroups,
    },
}
var listTagsbody = []model.Tag{
    {
        Key: "tag1",
        Value: "111",
    },
    {
```

```
            Key: "tag2",
            Value: "222",
        },
    }
    chargeInfobody := &model.ChargeInfo{
        ChargeMode: "postPaid",
    }
    logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
    mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordCreateClusterReqV2:= "your password"
    subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.CreateClusterReqV2{
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionCreateClusterReqV2,
        Tags: &listTagsbody,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
        NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components: "Storm,Kafka,Flume,Ranger",
        SubnetName: "subnet",
        SubnetId: &subnetIdCreateClusterReqV2,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "STREAMING",
        ClusterName: "mrs_Dokle_dm",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建一个混合集群，集群版本号为MRS 3.1.0。其中包含一个Master节点组，节点数为2；两个Core节点组，每个Core节点组的节点数均为3；两个Task节点组，一个Task节点组节点数为1，另一个节点数为0。

```
package main

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

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

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

```
client := mrs.NewMrsClient(
    mrs.MrsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateClusterRequest{}
dataVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups3 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups3 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeNodeGroups4 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups4 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeCountNodeGroups2:= int32(1)
dataVolumeCountNodeGroups3:= int32(1)
dataVolumeCountNodeGroups4:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(2),
        NodeSize: "Sit3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups4,
        DataVolume: dataVolumeNodeGroups4,
        DataVolumeCount: &dataVolumeCountNodeGroups,
    },
    {
        GroupName: "core_node_streaming_group",
        NodeNum: int32(3),
        NodeSize: "Sit3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups3,
        DataVolume: dataVolumeNodeGroups3,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
    },
```

```
        {
            GroupName: "core_node_analysis_group",
            NodeNum: int32(3),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups2,
            DataVolume: dataVolumeNodeGroups2,
            DataVolumeCount: &dataVolumeCountNodeGroups2,
        },
        {
            GroupName: "task_node_analysis_group",
            NodeNum: int32(1),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups1,
            DataVolume: dataVolumeNodeGroups1,
            DataVolumeCount: &dataVolumeCountNodeGroups3,
        },
        {
            GroupName: "task_node_streaming_group",
            NodeNum: int32(0),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups,
            DataVolume: dataVolumeNodeGroups,
            DataVolumeCount: &dataVolumeCountNodeGroups4,
        },
    }
    var listTagsbody = []model.Tag{
        {
            Key: "tag1",
            Value: "111",
        },
        {
            Key: "tag2",
            Value: "222",
        },
    }
    chargeInfobody := &model.ChargeInfo{
        ChargeMode: "postPaid",
    }
    logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
    mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordCreateClusterReqV2:= "your password"
    subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.CreateClusterReqV2{
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionCreateClusterReqV2,
        Tags: &listTagsbody,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
        NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components:
"Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        SubnetName: "subnet",
        SubnetId: &subnetIdCreateClusterReqV2,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "MIXED",
        ClusterName: "mrs_onmm_dm",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```

```
    }
}
```

- 创建自定义管控合设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为3；两个Core节点组，一个节点数为3，另一个节点数为1。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listAssignedRolesNodeGroups = []string{
        "NodeManager",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "FlinkResource",
    }
    dataVolumeNodeGroups := &model.Volume{
        Type: "SAS",
        Size: int32(600),
    }
    rootVolumeNodeGroups := &model.Volume{
        Type: "SAS",
        Size: int32(480),
    }
    var listAssignedRolesNodeGroups1 = []string{
        "DataNode",
        "NodeManager",
        "RegionServer",
        "Flume:1",
        "Broker",
        "Supervisor",
        "Logviewer",
        "HBaseIndexer",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "HSBroker:1,2",
        "ThriftServer",
        "ThriftServer1",
```

```
    "RESTServer",
    "FlinkResource",
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups2 = []string{
    "OMSServer:1,2",
    "SlapdServer:1,2",
    "KerberosServer:1,2",
    "KerberosAdmin:1,2",
    "quorumpeer:1,2,3",
    "NameNode:2,3",
    "Zkfc:2,3",
    "JournalNode:1,2,3",
    "ResourceManager:2,3",
    "JobHistoryServer:2,3",
    "DBServer:1,3",
    "Hue:1,3",
    "LoaderServer:1,3",
    "MetaStore:1,2,3",
    "WebHCat:1,2,3",
    "HiveServer:1,2,3",
    "HMaster:2,3",
    "MonitorServer:1,2",
    "Nimbus:1,2",
    "UI:1,2",
    "JDBCServer2x:1,2,3",
    "JobHistory2x:2,3",
    "SparkResource2x:1,2,3",
    "oozie:2,3",
    "LoadBalancer:2,3",
    "TezUI:1,3",
    "TimelineServer:3",
    "RangerAdmin:1,2",
    "UserSync:2",
    "TagSync:2",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSConsole:2,3",
    "FlinkResource:1,2,3",
    "DataNode:1,2,3",
    "NodeManager:1,2,3",
    "IndexServer2x:1,2",
    "ThriftServer:1,2,3",
    "RegionServer:1,2,3",
    "ThriftServer1:1,2,3",
    "RESTServer:1,2,3",
    "Broker:1,2,3",
    "Supervisor:1,2,3",
    "Logviewer:1,2,3",
    "Flume:1,2,3",
    "HSBroker:1,2,3",
}
dataVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
```

```
    dataVolumeCountNodeGroups1:= int32(1)
    dataVolumeCountNodeGroups2:= int32(1)
    var listNodeGroupsbody = []model.NodeGroupV2{
        {
            GroupName: "master_node_default_group",
            NodeNum: int32(3),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups2,
            DataVolume: dataVolumeNodeGroups2,
            DataVolumeCount: &dataVolumeCountNodeGroups,
            AssignedRoles: &listAssignedRolesNodeGroups2,
        },
        {
            GroupName: "node_group_1",
            NodeNum: int32(3),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups1,
            DataVolume: dataVolumeNodeGroups1,
            DataVolumeCount: &dataVolumeCountNodeGroups1,
            AssignedRoles: &listAssignedRolesNodeGroups1,
        },
        {
            GroupName: "node_group_2",
            NodeNum: int32(1),
            NodeSize: "Sit3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups,
            DataVolume: dataVolumeNodeGroups,
            DataVolumeCount: &dataVolumeCountNodeGroups2,
            AssignedRoles: &listAssignedRolesNodeGroups,
        },
    }
    var listTagsbody = []model.Tag{
        {
            Key: "tag1",
            Value: "111",
        },
        {
            Key: "tag2",
            Value: "222",
        },
    }
    chargeInfobody := &model.ChargeInfo{
        ChargeMode: "postPaid",
    }
    logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
    templateIdCreateClusterReqV2:= "mgmt_control_combined_v2"
    mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordCreateClusterReqV2:= "your password"
    subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.CreateClusterReqV2{
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionCreateClusterReqV2,
        Tags: &listTagsbody,
        TemplateId: &templateIdCreateClusterReqV2,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
        NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components:
"Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        SubnetName: "subnet",
        SubnetId: &subnetIdCreateClusterReqV2,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "CUSTOM",
        ClusterName: "mrs_heshe_dm",
```

```
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建自定义管控分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为5；一个Core节点组，节点数为3。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listAssignedRolesNodeGroups = []string{
        "DataNode",
        "NodeManager",
        "RegionServer",
        "Flume:1",
        "Broker",
        "Supervisor",
        "Logviewer",
        "HBaseIndexer",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "HSBroker:1,2",
        "ThriftServer",
        "ThriftServer1",
        "RESTServer",
        "FlinkResource",
    }
    dataVolumeNodeGroups := &model.Volume{
        Type: "SAS",
        Size: int32(600),
    }
    rootVolumeNodeGroups := &model.Volume{
```

```
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups1 = []string{
    "OMSServer:1,2",
    "SlapdServer:3,4",
    "KerberosServer:3,4",
    "KerberosAdmin:3,4",
    "quorumpeer:3,4,5",
    "NameNode:4,5",
    "Zkfc:4,5",
    "JournalNode:1,2,3,4,5",
    "ResourceManager:4,5",
    "JobHistoryServer:4,5",
    "DBServer:3,5",
    "Hue:1,2",
    "LoaderServer:1,2",
    "MetaStore:1,2,3,4,5",
    "WebHCat:1,2,3,4,5",
    "HiveServer:1,2,3,4,5",
    "HMaster:4,5",
    "MonitorServer:1,2",
    "Nimbus:1,2",
    "UI:1,2",
    "JDBCServer2x:1,2,3,4,5",
    "JobHistory2x:4,5",
    "SparkResource2x:1,2,3,4,5",
    "oozie:1,2",
    "LoadBalancer:1,2",
    "TezUI:1,2",
    "TimelineServer:5",
    "RangerAdmin:1,2",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSConsole:1,2",
    "FlinkResource:1,2,3,4,5",
    "DataNode:1,2,3,4,5",
    "NodeManager:1,2,3,4,5",
    "IndexServer2x:1,2",
    "ThriftServer:1,2,3,4,5",
    "RegionServer:1,2,3,4,5",
    "ThriftServer1:1,2,3,4,5",
    "RESTServer:1,2,3,4,5",
    "Broker:1,2,3,4,5",
    "Supervisor:1,2,3,4,5",
    "Logviewer:1,2,3,4,5",
    "Flume:1,2,3,4,5",
    "HBaseIndexer:1,2,3,4,5",
    "TagSync:1",
    "UserSync:1",
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(5),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups1,
        DataVolume: dataVolumeNodeGroups1,
```

```
            DataVolumeCount: &dataVolumeCountNodeGroups,
            AssignedRoles: &listAssignedRolesNodeGroups1,
        },
        {
            GroupName: "node_group_1",
            NodeNum: int32(3),
            NodeSize: "rc3.4xlarge.4.linux.bigdata",
            RootVolume: rootVolumeNodeGroups,
            DataVolume: dataVolumeNodeGroups,
            DataVolumeCount: &dataVolumeCountNodeGroups1,
            AssignedRoles: &listAssignedRolesNodeGroups,
        },
    }
    var listTagsbody = []model.Tag{
        {
            Key: "aaa",
            Value: "111",
        },
        {
            Key: "bbb",
            Value: "222",
        },
    }
    chargeInfobody := &model.ChargeInfo{
        ChargeMode: "postPaid",
    }
    logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
    templateIdCreateClusterReqV2:= "mgmt_control_separated_v2"
    mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordCreateClusterReqV2:= "your password"
    subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.CreateClusterReqV2{
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionCreateClusterReqV2,
        Tags: &listTagsbody,
        TemplateId: &templateIdCreateClusterReqV2,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
        NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components:
"Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        SubnetName: "subnet",
        SubnetId: &subnetIdCreateClusterReqV2,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "CUSTOM",
        ClusterName: "mrs_jdRU_dm01",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建自定义数据分设集群，集群版本号为MRS 3.1.0。包含一个Master节点组，节点数为9；四个Core节点组，每个Core节点组的节点数均为3。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listAssignedRolesNodeGroups = []string{
        "Broker",
        "Supervisor",
        "Logviewer",
        "KerberosClient",
        "SlapdClient",
        "meta",
    }
    dataVolumeNodeGroups := &model.Volume{
        Type: "SAS",
        Size: int32(600),
    }
    rootVolumeNodeGroups := &model.Volume{
        Type: "SAS",
        Size: int32(480),
    }
    var listAssignedRolesNodeGroups1 = []string{
        "KerberosClient",
        "SlapdClient",
        "meta",
    }
    dataVolumeNodeGroups1 := &model.Volume{
        Type: "SAS",
        Size: int32(600),
    }
    rootVolumeNodeGroups1 := &model.Volume{
        Type: "SAS",
        Size: int32(480),
    }
    var listAssignedRolesNodeGroups2 = []string{
        "HBaseIndexer",
        "SolrServer[3]",
        "EsNode[2]",
        "KerberosClient",
        "SlapdClient",
        "meta",
        "SolrServerAdmin:1,2",
    }
    dataVolumeNodeGroups2 := &model.Volume{
        Type: "SAS",
        Size: int32(600),
```

```
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups3 = []string{
    "DataNode",
    "NodeManager",
    "RegionServer",
    "Flume:1",
    "GraphServer",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSBroker:1,2",
}
dataVolumeNodeGroups3 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups3 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups4 = []string{
    "OMSServer:1,2",
    "SlapdServer:5,6",
    "KerberosServer:5,6",
    "KerberosAdmin:5,6",
    "quorumpeer:5,6,7,8,9",
    "NameNode:3,4",
    "Zkfc:3,4",
    "JournalNode:5,6,7",
    "ResourceManager:8,9",
    "JobHistoryServer:8",
    "DBServer:8,9",
    "Hue:8,9",
    "FlinkResource:3,4",
    "LoaderServer:3,5",
    "MetaStore:8,9",
    "WebHCat:5",
    "HiveServer:8,9",
    "HMaster:8,9",
    "FTP-Server:3,4",
    "MonitorServer:3,4",
    "Nimbus:8,9",
    "UI:8,9",
    "JDBCServer2x:8,9",
    "JobHistory2x:8,9",
    "SparkResource2x:5,6,7",
    "oozie:4,5",
    "EsMaster:7,8,9",
    "LoadBalancer:8,9",
    "TezUI:5,6",
    "TimelineServer:5",
    "RangerAdmin:4,5",
    "UserSync:5",
    "TagSync:5",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSBroker:5",
    "HSConsole:3,4",
    "FlinkResource:3,4",
}
dataVolumeNodeGroups4 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
```

```
rootVolumeNodeGroups4 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeCountNodeGroups2:= int32(1)
dataVolumeCountNodeGroups3:= int32(1)
dataVolumeCountNodeGroups4:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(9),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups4,
        DataVolume: dataVolumeNodeGroups4,
        DataVolumeCount: &dataVolumeCountNodeGroups,
        AssignedRoles: &listAssignedRolesNodeGroups4,
    },
    {
        GroupName: "node_group_1",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups3,
        DataVolume: dataVolumeNodeGroups3,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
        AssignedRoles: &listAssignedRolesNodeGroups3,
    },
    {
        GroupName: "node_group_2",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups2,
        DataVolume: dataVolumeNodeGroups2,
        DataVolumeCount: &dataVolumeCountNodeGroups2,
        AssignedRoles: &listAssignedRolesNodeGroups2,
    },
    {
        GroupName: "node_group_3",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups1,
        DataVolume: dataVolumeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups3,
        AssignedRoles: &listAssignedRolesNodeGroups1,
    },
    {
        GroupName: "node_group_4",
        NodeNum: int32(3),
        NodeSize: "rc3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups,
        DataVolume: dataVolumeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups4,
        AssignedRoles: &listAssignedRolesNodeGroups,
    },
}
var listTagsbody = []model.Tag{
    {
        Key: "aaa",
        Value: "111",
    },
    {
        Key: "bbb",
        Value: "222",
    },
}
chargeInfobody := &model.ChargeInfo{
    ChargeMode: "postPaid",
}
```

```
    logCollectionCreateClusterReqV2:= model.GetCreateClusterReqV2LogCollectionEnum().E_1
    templateIdCreateClusterReqV2:= "mgmt_control_data_separated_v2"
    mrsEcsDefaultAgencyCreateClusterReqV2:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordCreateClusterReqV2:= "your password"
    subnetIdCreateClusterReqV2:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.CreateClusterReqV2{
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionCreateClusterReqV2,
        Tags: &listTagsbody,
        TemplateId: &templateIdCreateClusterReqV2,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyCreateClusterReqV2,
        NodeRootPassword: &nodeRootPasswordCreateClusterReqV2,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components:
"Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        SubnetName: "subnet",
        SubnetId: &subnetIdCreateClusterReqV2,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "CUSTOM",
        ClusterName: "mrs_jdRU_dm02",
        ClusterVersion: "MRS 3.1.0",
    }
    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	正常响应示例。

错误码

请参见**错误码**。

6.1.2 修改集群名称

功能介绍

修改集群名称

接口约束

无

调用方法

请参见**如何调用API**。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/cluster-name

表 6-19 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-20 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_name	是	String	**参数解释：** 新的集群名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-21 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 更新映射请求操作结果。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed：操作失败 **默认取值：** 不涉及

状态码： 400

表 6-22 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

修改MRS集群名称为mrs_jdRU_dm01

```
{
  "cluster_name" : "mrs_jdRU_dm01"
}
```

响应示例

状态码： 200

修改集群名称成功

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改MRS集群名称为mrs_jdRU_dm01

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class UpdateClusterNameSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterNameRequest request = new UpdateClusterNameRequest();
        request.withClusterId("{cluster_id}");
        UpdateClusterReq body = new UpdateClusterReq();
        body.withClusterName("mrs_jdRU_dm01");
        request.withBody(body);
        try {
            UpdateClusterNameResponse response = client.updateClusterName(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改MRS集群名称为mrs_jdRU_dm01

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterNameRequest()
        request.cluster_id = "{cluster_id}"
        request.body = UpdateClusterReq(
            cluster_name="mrs_jdRU_dm01"
        )
        response = client.update_cluster_name(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

修改MRS集群名称为mrs_jdRU_dm01

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterNameRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.UpdateClusterReq{
        ClusterName: "mrs_jdRU_dm01",
    }
    response, err := client.UpdateClusterName(request)
    if err == nil {
```

```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	修改集群名称成功
400	修改集群名称失败

错误码

请参见**错误码**。

6.1.3 创建集群并提交作业

功能介绍

创建一个MRS集群并提交作业，并支持作业完成后删除集群，支持MRS 1.8.9及以上集群版本使用。使用接口前，您需要先获取下的资源信息。

- 通过VPC创建或查询VPC、子网
- 通过ECS创建或查询密钥对
- 通过**终端节点**获取区域信息
- 参考**MRS服务支持的组件**获取MRS版本及对应版本支持的组件信息

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/run-job-flow

表 6-23 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-24 请求 Body 参数

参数	是否必选	参数类型	描述
is_dec_project	否	Boolean	**参数解释：** 说明是否为专属云的资源。 **约束限制：** 不涉及 **取值范围：** ● true：是专属云的资源。 ● false：不是专属云的资源。 **默认取值：** false
cluster_version	是	String	**参数解释：** 集群版本。例如：MRS 3.1.0。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_name	是	String	**参数解释：** 集群名称。 **约束限制：** 不涉及 **取值范围：** ● 不允许相同。 ● 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_type	是	String	**参数解释：** 集群类型。 **约束限制：** 不涉及 **取值范围：** ● ANALYSIS：分析集群 ● STREAMING：流式集群 ● MIXED：混合集群 ● CUSTOM：自定义集群，仅MRS 3.x版本支持。 **默认取值：** 不涉及
charge_info	否	**ChargeInfo** object	**参数解释：** 计费类型信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
region	是	String	**参数解释：** 集群所在区域信息，请参见**终端节点**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpc_name	是	String	**参数解释：** 子网所在VPC名称。通过VPC管理控制台获取名称： 1. 登录VPC管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。在“虚拟私有云”页面的列表中即可获取VPC名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_id	否	String	**参数解释：** 子网ID。通过VPC管理控制台获取子网ID： 1. 登录VPC管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网。 4. 单击对应子网名称，获取“网络ID”。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_name	是	String	**参数解释：** 子网名称。通过VPC管理控制台获取子网名称： 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网，获取子网名称。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。当仅填写“subnet_name”一个参数且VPC下存在同名子网时，创建集群时以VPC平台第一个名称的子网为准。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及
components	是	String	**参数解释：** 组件名称列表，用逗号分隔。支持的组件请参见**获取MRS集群信息**页面的“MRS服务支持的组件”内容。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
external_data sources	否	Array of ClusterDataConnectorMap objects	**参数解释：** 部署Hive和Ranger等组件时，可以关联数据连接，将元数据存储于关联的数据库。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
availability_zone	是	String	**参数解释：** 可用分区名称，不支持多AZ集群。可用分区信息请参见**终端节点**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
security_groups_id	否	String	**参数解释：** 集群安全组的ID。 • 当该ID为空时MRS后台会自动创建安全组，自动创建的安全组名称以mrs_{cluster_name}开头。 • 当该ID不为空时，表示使用固定安全组来创建集群，传入的ID必须是当前租户中包含的安全组ID。 • 支持多个安全组ID，以逗号分隔。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
auto_create_default_security_group	否	Boolean	**参数解释：** 是否要创建MRS集群默认安全组。 **约束限制：** 当指定该参数为true，则无论“security_groups_id”参数是否指定，都会为集群创建默认安全组。 **取值范围：** ● true：创建MRS集群默认安全组。 ● false：不创建MRS集群默认安全组。 **默认取值：** false
safe_mode	是	String	**参数解释：** MRS集群运行模式。 **约束限制：** 不涉及 **取值范围：** ● SIMPLE：普通集群，表示Kerberos认证关闭，用户可使用集群提供的所有功能。 ● KERBEROS：安全集群，表示Kerberos认证开启，普通用户无权限使用MRS集群的“文件管理”和“作业管理”功能，并且无法查看Hadoop、Spark的作业记录以及集群资源使用情况。如果需要使用集群更多功能，需要找Manager的管理员分配权限。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
manager_admin_password	是	String	**参数解释：** 配置Manager管理员用户的密码。 **约束限制：** 不涉及 **取值范围：** ● 密码长度应在8～26个字符之间。 ● 至少包含四种字符组合，如大写字母，小写字母，数字，特殊字符（!@$%^-_=+[{}]:,./?），但不能包含空格。 ● 不能与用户名或者倒序用户名相同。 **默认取值：** 不涉及
login_mode	是	String	**参数解释：** 节点登录方式。 **约束限制：** 不涉及 **取值范围：** ● PASSWORD：密码登录，选择此项时，node_root_password不能为空。 ● KEYPAIR：密钥对登录，选择此项时，node_keypair_name不能为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_root_password	否	String	**参数解释：** 配置访问集群节点的root密码。 **约束限制：** 不涉及 **取值范围：** • 字符串类型，可输入的字符串长度为8-26。 • 至少包含四种字符组合，如大写字母，小写字母，数字，特殊字符（!@$%^-_=+[{}]:,./?），但不能包含空格。 • 不能与用户名或者倒序用户名相同。 **默认取值：** 不涉及
node_keypair_name	否	String	**参数解释：** 密钥对名称。用户可以使用密钥对方式登录集群节点。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
enterprise_project_id	否	String	**参数解释：** 企业项目ID。创建集群时，给集群绑定企业项目ID。获取方式请参见《企业管理API参考》的“查询企业项目列表”响应消息表“enterprise_project字段数据结构说明”的“id”,即**表5 enterprise_project_list字段数据结构说明**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 默认设置为0，表示为default企业项目。

参数	是否必选	参数类型	描述
eip_address	否	String	**参数解释：** 与MRS集群绑定的弹性公网IP，可实现使用弹性公网IP访问Manager的目的。该弹性公网IP必须已经创建且与集群在同一区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eip_id	否	String	**参数解释：** 绑定的弹性公网IP的ID。 **约束限制：** 当“eip_address”配置时，该参数必须配置，用于表示绑定的弹性公网IP的ID。可通过在VPC服务的“网络控制台 > 弹性公网IP和带宽 > 弹性公网IP”页面单击待绑定的弹性公网IP，在基本信息中获取“ID”。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
mrs_ecs_default_agency	否	String	**参数解释：** 集群节点默认绑定的委托名称，固定为MRS_ECS_DEFAULT_AGENCY。通过绑定委托，您可以将部分资源共享给ECS或BMS云服务来管理，例如通过配置ECS委托可自动获取AK/SK访问OBS。MRS_ECS_DEFAULT_AGENCY委托拥有对象存储服务的OBS OperateAccess权限和在集群所在区域拥有CES FullAccess（对开启细粒度策略的用户）、CES Administrator和KMS Administrator权限。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
template_id	否	String	**参数解释：** 当集群类型为CUSTOM时，用于指定节点部署所使用的模板。 • mgmt_control_combined_v2：管控合设模板，管理角色和控制角色共同部署在Master节点中，数据实例合设在同一节点组。该部署方式适用于100个以下的节点，可以减少成本。 • mgmt_control_separated_v2：管控分设模板，管理角色和控制角色分别部署在不同的Master节点中，数据实例合设在同一节点组。该部署方式适用于100-500个节点，在高并发负载情况下表现更好。 • mgmt_control_data_separated_v2：数据分设模板，管理角色和控制角色分别部署在不同的Master节点中，数据实例分设在不同节点组。该部署方式适用于500个以上的节点，可以将各组件进一步分开部署，适用于更大的集群规模。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	否	Array of **Tag** objects	**参数解释：** 集群的标签信息。 **约束限制：** 同一个集群最多能使用20个tag，tag的名称（key）不能重复。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
log_collection	否	Integer	**参数解释：** 集群创建失败时，是否收集失败日志。 **约束限制：** 不涉及 **取值范围：** ● 0：不创建OBS桶仅用于MRS集群创建失败时的日志收集。 ● 1：创建OBS桶仅用于MRS集群创建失败时的日志收集。 **默认取值：** 1
node_groups	是	Array of **NodeGroupV2** objects	**参数解释：** 组成集群的节点组信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
bootstrap_scripts	否	Array of **BootstrapScript** objects	**参数解释：** 配置引导操作脚本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
log_uri	否	String	**参数解释：** 集群日志转储至OBS的具体路径。 开启日志转储功能后，日志上传需要对应OBS路径的读写权限，请配置MRS_ECS_DEFULT_AGENCY默认委托或具有对应OBS路径读写权限的自定义委托。具体请参见**配置存算分离集群（委托方式）**。该参数只适用于支持“集群日志转储OBS”特性的集群版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
component_configs	否	Array of **ComponentConfig** objects	**参数解释：** 集群组件自定义配置。该参数只适用于支持“自定义组件配置创建集群”特性的集群版本。 **约束限制：** 不能超过50条。 **取值范围：** 不涉及 **默认取值：** 不涉及
delete_when_no_steps	否	Boolean	**参数解释：** 作业完成后是否自动删除集群。 **约束限制：** 不涉及 **取值范围：** • true：作业完成后自动删除集群。 • false：作业完成后不自动删除集群。 **默认取值：** false。

参数	是否必选	参数类型	描述
steps	是	Array of **StepConfig** objects	**参数解释：** 作业列表。 **约束限制：** 不能超过255条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-25 ClusterDataConnectorMap

参数	是否必选	参数类型	描述
map_id	否	Integer	**参数解释：** 数据连接关联ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
connector_id	否	String	**参数解释：** 数据连接ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
component_name	否	String	**参数解释：** 组件名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
role_type	否	String	**参数解释：** 组件角色类型。 **约束限制：** 不涉及 **取值范围：** • hive_metastore：Hive Metastore角色 • hive_data：Hive角色 • hbase_data：Hbase角色 • ranger_data：Ranger角色 **默认取值：** 不涉及
source_type	否	String	**参数解释：** 数据连接类型。 **约束限制：** 不涉及 **取值范围：** • LOCAL_DB：本地元数据 • RDS_POSTGRES：RDS服务PostgreSQL数据库 • RDS_MYSQL：RDS服务MySQL数据库 • gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及
cluster_id	否	String	**参数解释：** 关联集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
status	否	Integer	**参数解释：** 数据连接状态。 **约束限制：** 不涉及 **取值范围：** ● 0：代表正常状态 ● 1：代表使用中 **默认取值：** 不涉及

表 6-26 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 同一资源的key值不能重复。 ● 最大长度128个unicode字符，不能为空字符串。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 最大长度255个unicode字符，可以为空字符串。 **默认取值：** 不涉及

表 6-27 NodeGroupV2

参数	是否必选	参数类型	描述
group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”组成，且长度为[1-64]个字符。 节点组配置原则如下： ● master_node_default_group：Master节点组，所有集群类型均需包含该节点组。 ● core_node_analysis_group：分析Core节点组，分析集群、混合集群均需包含该节点组。 ● core_node_streaming_group：流式Core节点组，流式集群和混合集群均需包含该节点组。 ● task_node_analysis_group：分析Task节点组，分析集群和混合集群可根据需要选择该节点组。 ● task_node_streaming_group：流式Task节点组，流式集群、混合集群可根据需要选择该节点组。 ● node_group{x}：自定义集群节点组，可根据需要添加多个，最多支持添加9个该节点组。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_num	是	Integer	**参数解释：** 节点数量。 **约束限制：** Core与Task节点总数最大为500个。 **取值范围：** 0-500 **默认取值：** 不涉及
node_size	是	String	**参数解释：** 节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume	否	**Volume** object	**参数解释：** 节点系统盘信息，部分虚拟机或BMS自带系统盘的情况该参数可选，其他情况该参数必选。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_volume	否	**Volume** object	**参数解释：** 节点数据盘信息。 **约束限制：** 当data_volume_count不为0时，该参数必选。 **取值范围：** 不涉及 **默认取值：** 不涉及
data_volume_count	否	Integer	**参数解释：** 节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 0-20 **默认取值：** 不涉及
charge_info	否	**ChargeInfo** object	**参数解释：** 节点组的计费类型，Master和Core节点组是和集群的计费类型一致，Task节点组可以和集群的计费类型不同。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
auto_scaling_policy	否	**AutoScalingPolicy** object	**参数解释：** 弹性伸缩规则信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
assigned_roles	否	Array of strings	**参数解释：** 当集群类型为CUSTOM时，该参数必选。可以指定节点组中部署的角色，该参数是一个字符串数组，每个字符串表示一个角色表达式。 角色表达式定义： ● 当该角色在节点组所有节点部署时： {role name}，如“DataNode”。 ● 当该角色在节点组指定下标节点部署时：{role name}:{index1},{index2}…,{indexN}，如“NameNode:1,2”，下标从1开始计数。 可选的角色请参考**MRS支持的角色与组件对应表**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-28 Volume

参数	是否必选	参数类型	描述
type	是	String	**参数解释：** 磁盘类型。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO磁盘类型。 ● SAS：高IO磁盘类型。 ● SSD：超高IO磁盘类型。 ● GPSSD：通用型SSD磁盘类型 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
size	是	Integer	**参数解释：** 数据盘大小，容量单位为GB。 **约束限制：** 不涉及 **取值范围：** 10-32768 **默认取值：** 不涉及

表 6-29 ChargeInfo

参数	是否必选	参数类型	描述
charge_mode	是	String	**参数解释：** 计费模式。 **约束限制：** 不涉及 **取值范围：** ● prePaid：预付费，即包年/包月。（创建集群接口现已支持预付费，创建集群并提交作业接口暂不支持预付费。） ● postPaid：后付费，即按需计费。 **默认取值：** 不涉及
period_type	否	String	**参数解释：** 周期类型。 **约束限制：** 不涉及 **取值范围：** ● month：包月。 ● year： 包年。 ● day：按需计费。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
period_num	否	Integer	**参数解释：** 周期数。 **约束限制：** “charge_mode”为“prePaid”时生效，且为必选值，指定订购的时间。 **取值范围：** ● 当“period_type”为“month”时，取值为1-9。 ● 当“period_type”为“year”时，取值为1-3。 **默认取值：** 不涉及
is_auto_pay	否	Boolean	**参数解释：** 是否自动支付，包周期模式下使用，下单订购后，是否自动从客户的账户中支付，而不需要客户手动去进行支付。 **约束限制：** 不涉及 **取值范围：** ● true：自动支付，会自动选择折扣和优惠券进行优惠，然后自动从客户账户中支付，自动支付失败后会生成订单成功、但订单状态为“待支付”，等待客户手动支付。 ● false：手动支付，需要客户手动去支付，客户可以选择折扣和优惠券。 **默认取值：** false

表 6-30 AutoScalingPolicy

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** ● true：开启自动伸缩规则 ● false：不开启自动伸缩规则 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
exec_scripts	否	Array of **ScaleScript** objects	**参数解释：** 弹性伸缩自定义自动化脚本列表。若该参数为空表示不启用自动化脚本。在V2弹性伸缩策略创建和更新接口中暂时不支持该字段。 **约束限制：** 不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-31 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	**参数解释：** 资源计划的起始时间，格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
effective_days	否	Array of strings	**参数解释：** 资源计划的生效日期，为空时代表每日，另外也可为以下返回值： MONDAY（周一）、TUESDAY（周二）、WEDNESDAY（周三）、THURSDAY（周四）、FRIDAY（周五）、SATURDAY（周六）、SUNDAY（周日） **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-32 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 在一个节点组范围内，不允许重名。 **默认取值：** 不涉及
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度为[0-1024]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** • scale_out：扩容 • scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-33 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。详细指标名称内容请参见"**弹性伸缩指标列表**"。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见"**弹性伸缩指标列表**"。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** • LT：小于 • GT：大于 • LTOE：小于等于 • GTOE：大于等于 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-288 **默认取值：** 不涉及

表 6-34 ScaleScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩自定义自动化脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的自定义自动化脚本名称不允许相同。 只能由英文字母、数字、空格以及“_”和“-”组成，不能以空格开头，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 自定义自动化脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。示例：obs://XXX/scale.sh ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 自定义自动化脚本参数。多个参数间用空格隔开。 可以传入以下系统预定义参数： ● ${mrs_scale_node_num}：扩缩容节点数 ● ${mrs_scale_type}：扩缩容类型，扩容为scale_out，缩容为scale_in ● ${mrs_scale_node_hostnames}：扩缩容的节点主机名称 ● ${mrs_scale_node_ips}：扩缩容的节点IP ● ${mrs_scale_rule_name}：触发扩缩容的规则名 其他用户自定义参数使用方式与普通shell脚本相同，多个参数中间用空格隔开。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 自定义自动化脚本所执行的节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 自定义自动化脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：自定义自动化脚本只运行在主Master节点上。 ● false：自定义自动化脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	是	String	**参数解释：** 自定义自动化脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此自定义自动化脚本是否执行成功，则集群都能继续安装和启动。 **约束限制：** 由于缩容成功无法回滚，因此缩容后执行的脚本“fail_action”必须设置为“continue”。 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
action_stage	是	String	**参数解释：** 脚本执行时机。 **约束限制：** 不涉及 **取值范围：** ● before_scale_out：扩容前 ● before_scale_in：缩容前 ● after_scale_out：扩容后 ● after_scale_in：缩容后 **默认取值：** 不涉及

表 6-35 BootstrapScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 引导操作脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的引导操作脚本名称不允许相同。 只能由英文字母、数字、空格以及“_”和“-”组成，不能以空格开头，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 引导操作脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 OBS桶的路径：直接手动输入脚本路径。例如输入MRS提供的公共样例脚本路径。示例：obs://bootstrap/presto/presto-install.sh，其中安装dualroles时，presto-install.sh脚本参数为dualroles, 安装worker时，presto-install.sh脚本参数为worker。根据Presto使用习惯，建议您在Active Master节点上安装dualroles，在Core节点上安装worker。 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 引导操作脚本参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 引导操作脚本所执行的节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 引导操作脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本只运行在主Master节点上。 ● false：引导操作脚本可运行在所有Master节点上。 **默认取值：** 不涉及
fail_action	是	String	**参数解释：** 引导操作脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“继续”，无论此引导操作是否执行成功，则集群都能继续安装和启动。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
before_component_start	否	Boolean	**参数解释：** 引导操作脚本执行的时间。目前支持“组件启动前”和“组件启动后”两种类型。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本在组件启动前执行。 ● false：引导操作脚本在组件启动后执行。 **默认取值：** false
start_time	否	Long	**参数解释：** 单个引导操作脚本的执行时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
state	否	String	**参数解释：** 单个引导操作脚本的运行状态。 **约束限制：** 不涉及 **取值范围：** ● PENDING：挂起 ● IN_PROGRESS：处理中 ● SUCCESS：处理成功 ● FAILURE：处理失败 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
action_stages	否	Array of strings	**参数解释：** 选择引导操作脚本执行的时间。 **约束限制：** 参数枚举值： • BEFORE_COMPONENT_FIRST_START: 组件首次启动前 • AFTER_COMPONENT_FIRST_START: 组件首次启动后 • BEFORE_SCALE_IN: 缩容前 • AFTER_SCALE_IN: 缩容后 • BEFORE_SCALE_OUT: 扩容前 • AFTER_SCALE_OUT: 扩容后 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-36 ComponentConfig

参数	是否必选	参数类型	描述
component_name	是	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
configs	否	Array of **Config** objects	**参数解释：** 组件配置项列表。 **约束限制：** 不能超过100条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-37 Config

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 配置名，仅支持MRS组件配置页面上所展示的配置名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
value	是	String	**参数解释：** 配置值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
config_file_name	是	String	**参数解释：** 配置文件名，仅支持MRS组件配置页面上所展示的文件名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-38 StepConfig

参数	是否必选	参数类型	描述
job_execution	是	JobExecution object	**参数解释：** 作业参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-39 JobExecution

参数	是否必选	参数类型	描述
job_type	是	String	**参数解释：** 作业类型。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● SparkSubmit ● SparkPython：该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。 ● HiveScript ● HiveSql ● DistCp，导入、导出数据。 ● SparkScript ● SparkSql ● Flink **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_name	是	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及
arguments	否	Array of strings	**参数解释：** 程序执行的关键参数，该参数由用户程序内的函数指定，MRS只负责参数的传入。 **约束限制：** 参数最多为150000字符，不能包含;|&>'<$!'"\特殊字符，可为空。 说明： • 若输入带有敏感信息（如登录密码）的参数可能在作业详情展示和日志打印中存在暴露的风险，请谨慎操作。 • 提交HiveScript或HiveSql类型的作业时如需以“obs://”开头格式访问存储在OBS上的文件，请在Hive服务配置页面搜索参数“core.site.customized.configs”，新增OBS的endpoint配置项，参数为“fs.obs.endpoint”，值请输入OBS对应的endpoint，具体请参考**终端节点**。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
properties	否	Map<String,String>	**参数解释：** 程序系统参数。 **约束限制：** 参数最多为2048字符，不能包含><|\`&!\特殊字符，可为空。 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-40 响应 Body 参数

参数	参数类型	描述
cluster_id	String	**参数解释：** 集群创建成功后系统返回的集群ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

创建一个版本号为MRS 3.1.0的自定义管控合设集群，并提交一个作业类型为HiveScript的作业。

```
POST /v2/{project_id}/run-job-flow

{
  "cluster_version" : "MRS 3.1.0",
  "cluster_name" : "mrs_heshe_dm",
  "cluster_type" : "CUSTOM",
  "charge_info" : {
    "charge_mode" : "postPaid"
  },
  "region" : "",
  "availability_zone" : "",
  "vpc_name" : "vpc-37cd",
  "subnet_id" : "1f8c5ca6-1f66-4096-bb00-baf175954f6e",
  "subnet_name" : "subnet",
  "components" : "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
```

```
  "safe_mode" : "KERBEROS",
  "manager_admin_password" : "your password",
  "login_mode" : "PASSWORD",
  "node_root_password" : "your password",
  "mrs_ecs_default_agency" : "MRS_ECS_DEFAULT_AGENCY",
  "template_id" : "mgmt_control_combined_v2",
  "log_collection" : 1,
  "tags" : [ {
    "key" : "tag1",
    "value" : "111"
  }, {
    "key" : "tag2",
    "value" : "222"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 3,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "OMSServer:1,2", "SlapdServer:1,2", "KerberosServer:1,2", "KerberosAdmin:1,2",
"quorumpeer:1,2,3", "NameNode:2,3", "Zkfc:2,3", "JournalNode:1,2,3", "ResourceManager:2,3",
"JobHistoryServer:2,3", "DBServer:1,3", "Hue:1,3", "LoaderServer:1,3", "MetaStore:1,2,3", "WebHCat:1,2,3",
"HiveServer:1,2,3", "HMaster:2,3", "MonitorServer:1,2", "Nimbus:1,2", "UI:1,2", "JDBCServer2x:1,2,3",
"JobHistory2x:2,3", "SparkResource2x:1,2,3", "oozie:2,3", "LoadBalancer:2,3", "TezUI:1,3", "TimelineServer:3",
"RangerAdmin:1,2", "UserSync:2", "TagSync:2", "KerberosClient", "SlapdClient", "meta", "HSConsole:2,3",
"FlinkResource:1,2,3", "DataNode:1,2,3", "NodeManager:1,2,3", "IndexServer2x:1,2", "ThriftServer:1,2,3",
"RegionServer:1,2,3", "ThriftServer1:1,2,3", "RESTServer:1,2,3", "Broker:1,2,3", "Supervisor:1,2,3",
"Logviewer:1,2,3", "Flume:1,2,3", "HSBroker:1,2,3" ]
  }, {
    "group_name" : "node_group_1",
    "node_num" : 3,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "DataNode", "NodeManager", "RegionServer", "Flume:1", "Broker", "Supervisor",
"Logviewer", "HBaseIndexer", "KerberosClient", "SlapdClient", "meta", "HSBroker:1,2", "ThriftServer",
"ThriftServer1", "RESTServer", "FlinkResource" ]
  }, {
    "group_name" : "node_group_2",
    "node_num" : 1,
    "node_size" : "Sit3.4xlarge.4.linux.bigdata",
    "root_volume" : {
      "type" : "SAS",
      "size" : 480
    },
    "data_volume" : {
      "type" : "SAS",
      "size" : 600
    },
    "data_volume_count" : 1,
    "assigned_roles" : [ "NodeManager", "KerberosClient", "SlapdClient", "meta", "FlinkResource" ]
  } ],
  "log_uri" : "obs://bucketTest/logs",
  "delete_when_no_steps" : true,
```

```
 "steps" : [ {
   "job_execution" : {
     "job_name" : "import_file",
     "job_type" : "DistCp",
     "arguments" : [ "obs://test/test.sql", "/user/hive/input" ]
   }
 }, {
   "job_execution" : {
     "job_name" : "hive_test",
     "job_type" : "HiveScript",
     "arguments" : [ "obs://test/hive/sql/HiveScript.sql" ]
   }
 } ]
}
```

响应示例

状态码： 200

正常响应示例。

```
{
  "cluster_id" : "da1592c2-bb7e-468d-9ac9-83246e95447a"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个版本号为MRS 3.1.0的自定义管控合设集群，并提交一个作业类型为HiveScript的作业。

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

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

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

public class RunJobFlowSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
        .build();
RunJobFlowRequest request = new RunJobFlowRequest();
RunJobFlowCommand body = new RunJobFlowCommand();
List<String> listJobExecutionArguments = new ArrayList<>();
listJobExecutionArguments.add("obs://test/hive/sql/HiveScript.sql");
JobExecution jobExecutionSteps = new JobExecution();
jobExecutionSteps.withJobType("HiveScript")
    .withJobName("hive_test")
    .withArguments(listJobExecutionArguments);
List<String> listJobExecutionArguments1 = new ArrayList<>();
listJobExecutionArguments1.add("obs://test/test.sql");
listJobExecutionArguments1.add("/user/hive/input");
JobExecution jobExecutionSteps1 = new JobExecution();
jobExecutionSteps1.withJobType("DistCp")
    .withJobName("import_file")
    .withArguments(listJobExecutionArguments1);
List<StepConfig> listbodySteps = new ArrayList<>();
listbodySteps.add(
    new StepConfig()
        .withJobExecution(jobExecutionSteps1)
);
listbodySteps.add(
    new StepConfig()
        .withJobExecution(jobExecutionSteps)
);
List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
listNodeGroupsAssignedRoles.add("NodeManager");
listNodeGroupsAssignedRoles.add("KerberosClient");
listNodeGroupsAssignedRoles.add("SlapdClient");
listNodeGroupsAssignedRoles.add("meta");
listNodeGroupsAssignedRoles.add("FlinkResource");
Volume dataVolumeNodeGroups = new Volume();
dataVolumeNodeGroups.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups = new Volume();
rootVolumeNodeGroups.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles1 = new ArrayList<>();
listNodeGroupsAssignedRoles1.add("DataNode");
listNodeGroupsAssignedRoles1.add("NodeManager");
listNodeGroupsAssignedRoles1.add("RegionServer");
listNodeGroupsAssignedRoles1.add("Flume:1");
listNodeGroupsAssignedRoles1.add("Broker");
listNodeGroupsAssignedRoles1.add("Supervisor");
listNodeGroupsAssignedRoles1.add("Logviewer");
listNodeGroupsAssignedRoles1.add("HBaseIndexer");
listNodeGroupsAssignedRoles1.add("KerberosClient");
listNodeGroupsAssignedRoles1.add("SlapdClient");
listNodeGroupsAssignedRoles1.add("meta");
listNodeGroupsAssignedRoles1.add("HSBroker:1,2");
listNodeGroupsAssignedRoles1.add("ThriftServer");
listNodeGroupsAssignedRoles1.add("ThriftServer1");
listNodeGroupsAssignedRoles1.add("RESTServer");
listNodeGroupsAssignedRoles1.add("FlinkResource");
Volume dataVolumeNodeGroups1 = new Volume();
dataVolumeNodeGroups1.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups1 = new Volume();
rootVolumeNodeGroups1.withType("SAS")
    .withSize(480);
List<String> listNodeGroupsAssignedRoles2 = new ArrayList<>();
listNodeGroupsAssignedRoles2.add("OMSServer:1,2");
listNodeGroupsAssignedRoles2.add("SlapdServer:1,2");
listNodeGroupsAssignedRoles2.add("KerberosServer:1,2");
listNodeGroupsAssignedRoles2.add("KerberosAdmin:1,2");
listNodeGroupsAssignedRoles2.add("quorumpeer:1,2,3");
listNodeGroupsAssignedRoles2.add("NameNode:2,3");
```

```
listNodeGroupsAssignedRoles2.add("Zkfc:2,3");
listNodeGroupsAssignedRoles2.add("JournalNode:1,2,3");
listNodeGroupsAssignedRoles2.add("ResourceManager:2,3");
listNodeGroupsAssignedRoles2.add("JobHistoryServer:2,3");
listNodeGroupsAssignedRoles2.add("DBServer:1,3");
listNodeGroupsAssignedRoles2.add("Hue:1,3");
listNodeGroupsAssignedRoles2.add("LoaderServer:1,3");
listNodeGroupsAssignedRoles2.add("MetaStore:1,2,3");
listNodeGroupsAssignedRoles2.add("WebHCat:1,2,3");
listNodeGroupsAssignedRoles2.add("HiveServer:1,2,3");
listNodeGroupsAssignedRoles2.add("HMaster:2,3");
listNodeGroupsAssignedRoles2.add("MonitorServer:1,2");
listNodeGroupsAssignedRoles2.add("Nimbus:1,2");
listNodeGroupsAssignedRoles2.add("UI:1,2");
listNodeGroupsAssignedRoles2.add("JDBCServer2x:1,2,3");
listNodeGroupsAssignedRoles2.add("JobHistory2x:2,3");
listNodeGroupsAssignedRoles2.add("SparkResource2x:1,2,3");
listNodeGroupsAssignedRoles2.add("oozie:2,3");
listNodeGroupsAssignedRoles2.add("LoadBalancer:2,3");
listNodeGroupsAssignedRoles2.add("TezUI:1,3");
listNodeGroupsAssignedRoles2.add("TimelineServer:3");
listNodeGroupsAssignedRoles2.add("RangerAdmin:1,2");
listNodeGroupsAssignedRoles2.add("UserSync:2");
listNodeGroupsAssignedRoles2.add("TagSync:2");
listNodeGroupsAssignedRoles2.add("KerberosClient");
listNodeGroupsAssignedRoles2.add("SlapdClient");
listNodeGroupsAssignedRoles2.add("meta");
listNodeGroupsAssignedRoles2.add("HSConsole:2,3");
listNodeGroupsAssignedRoles2.add("FlinkResource:1,2,3");
listNodeGroupsAssignedRoles2.add("DataNode:1,2,3");
listNodeGroupsAssignedRoles2.add("NodeManager:1,2,3");
listNodeGroupsAssignedRoles2.add("IndexServer2x:1,2");
listNodeGroupsAssignedRoles2.add("ThriftServer:1,2,3");
listNodeGroupsAssignedRoles2.add("RegionServer:1,2,3");
listNodeGroupsAssignedRoles2.add("ThriftServer1:1,2,3");
listNodeGroupsAssignedRoles2.add("RESTServer:1,2,3");
listNodeGroupsAssignedRoles2.add("Broker:1,2,3");
listNodeGroupsAssignedRoles2.add("Supervisor:1,2,3");
listNodeGroupsAssignedRoles2.add("Logviewer:1,2,3");
listNodeGroupsAssignedRoles2.add("Flume:1,2,3");
listNodeGroupsAssignedRoles2.add("HSBroker:1,2,3");
Volume dataVolumeNodeGroups2 = new Volume();
dataVolumeNodeGroups2.withType("SAS")
    .withSize(600);
Volume rootVolumeNodeGroups2 = new Volume();
rootVolumeNodeGroups2.withType("SAS")
    .withSize(480);
List<NodeGroupV2> listbodyNodeGroups = new ArrayList<>();
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("master_node_default_group")
        .withNodeNum(3)
        .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups2)
        .withDataVolume(dataVolumeNodeGroups2)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles2)
);
listbodyNodeGroups.add(
    new NodeGroupV2()
        .withGroupName("node_group_1")
        .withNodeNum(3)
        .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
        .withRootVolume(rootVolumeNodeGroups1)
        .withDataVolume(dataVolumeNodeGroups1)
        .withDataVolumeCount(1)
        .withAssignedRoles(listNodeGroupsAssignedRoles1)
);
listbodyNodeGroups.add(
```

```
            new NodeGroupV2()
                .withGroupName("node_group_2")
                .withNodeNum(1)
                .withNodeSize("Sit3.4xlarge.4.linux.bigdata")
                .withRootVolume(rootVolumeNodeGroups)
                .withDataVolume(dataVolumeNodeGroups)
                .withDataVolumeCount(1)
                .withAssignedRoles(listNodeGroupsAssignedRoles)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("tag1")
                .withValue("111")
        );
        listbodyTags.add(
            new Tag()
                .withKey("tag2")
                .withValue("222")
        );
        ChargeInfo chargeInfobody = new ChargeInfo();
        chargeInfobody.withChargeMode("postPaid");
        body.withSteps(listbodySteps);
        body.withDeleteWhenNoSteps(true);
        body.withLogUri("obs://bucketTest/logs");
        body.withNodeGroups(listbodyNodeGroups);
        body.withLogCollection(RunJobFlowCommand.LogCollectionEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withTemplateId("mgmt_control_combined_v2");
        body.withMrsEcsDefaultAgency("MRS_ECS_DEFAULT_AGENCY");
        body.withNodeRootPassword("your password");
        body.withLoginMode("PASSWORD");
        body.withManagerAdminPassword("your password");
        body.withSafeMode("KERBEROS");
        body.withAvailabilityZone("");

body.withComponents("Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez");
        body.withSubnetName("subnet");
        body.withSubnetId("1f8c5ca6-1f66-4096-bb00-baf175954f6e");
        body.withVpcName("vpc-37cd");
        body.withRegion("");
        body.withChargeInfo(chargeInfobody);
        body.withClusterType("CUSTOM");
        body.withClusterName("mrs_heshe_dm");
        body.withClusterVersion("MRS 3.1.0");
        request.withBody(body);
        try {
            RunJobFlowResponse response = client.runJobFlow(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建一个版本号为MRS 3.1.0的自定义管控合设集群，并提交一个作业类型为HiveScript的作业。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = RunJobFlowRequest()
        listArgumentsJobExecution = [
            "obs://test/hive/sql/HiveScript.sql"
        ]
        jobExecutionSteps = JobExecution(
            job_type="HiveScript",
            job_name="hive_test",
            arguments=listArgumentsJobExecution
        )
        listArgumentsJobExecution1 = [
            "obs://test/test.sql",
            "/user/hive/input"
        ]
        jobExecutionSteps1 = JobExecution(
            job_type="DistCp",
            job_name="import_file",
            arguments=listArgumentsJobExecution1
        )
        listStepsbody = [
            StepConfig(
                job_execution=jobExecutionSteps1
            ),
            StepConfig(
                job_execution=jobExecutionSteps
            )
        ]
        listAssignedRolesNodeGroups = [
            "NodeManager",
            "KerberosClient",
            "SlapdClient",
            "meta",
            "FlinkResource"
        ]
        dataVolumeNodeGroups = Volume(
            type="SAS",
            size=600
        )
        rootVolumeNodeGroups = Volume(
            type="SAS",
            size=480
        )
        listAssignedRolesNodeGroups1 = [
            "DataNode",
```

```
    "NodeManager",
    "RegionServer",
    "Flume:1",
    "Broker",
    "Supervisor",
    "Logviewer",
    "HBaseIndexer",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSBroker:1,2",
    "ThriftServer",
    "ThriftServer1",
    "RESTServer",
    "FlinkResource"
]
dataVolumeNodeGroups1 = Volume(
    type="SAS",
    size=600
)
rootVolumeNodeGroups1 = Volume(
    type="SAS",
    size=480
)
listAssignedRolesNodeGroups2 = [
    "OMSServer:1,2",
    "SlapdServer:1,2",
    "KerberosServer:1,2",
    "KerberosAdmin:1,2",
    "quorumpeer:1,2,3",
    "NameNode:2,3",
    "Zkfc:2,3",
    "JournalNode:1,2,3",
    "ResourceManager:2,3",
    "JobHistoryServer:2,3",
    "DBServer:1,3",
    "Hue:1,3",
    "LoaderServer:1,3",
    "MetaStore:1,2,3",
    "WebHCat:1,2,3",
    "HiveServer:1,2,3",
    "HMaster:2,3",
    "MonitorServer:1,2",
    "Nimbus:1,2",
    "UI:1,2",
    "JDBCServer2x:1,2,3",
    "JobHistory2x:2,3",
    "SparkResource2x:1,2,3",
    "oozie:2,3",
    "LoadBalancer:2,3",
    "TezUI:1,3",
    "TimelineServer:3",
    "RangerAdmin:1,2",
    "UserSync:2",
    "TagSync:2",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSConsole:2,3",
    "FlinkResource:1,2,3",
    "DataNode:1,2,3",
    "NodeManager:1,2,3",
    "IndexServer2x:1,2",
    "ThriftServer:1,2,3",
    "RegionServer:1,2,3",
    "ThriftServer1:1,2,3",
    "RESTServer:1,2,3",
    "Broker:1,2,3",
    "Supervisor:1,2,3",
```

```
        "Logviewer:1,2,3",
        "Flume:1,2,3",
        "HSBroker:1,2,3"
    ]
    dataVolumeNodeGroups2 = Volume(
        type="SAS",
        size=600
    )
    rootVolumeNodeGroups2 = Volume(
        type="SAS",
        size=480
    )
    listNodeGroupsbody = [
        NodeGroupV2(
            group_name="master_node_default_group",
            node_num=3,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups2,
            data_volume=dataVolumeNodeGroups2,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups2
        ),
        NodeGroupV2(
            group_name="node_group_1",
            node_num=3,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups1,
            data_volume=dataVolumeNodeGroups1,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups1
        ),
        NodeGroupV2(
            group_name="node_group_2",
            node_num=1,
            node_size="Sit3.4xlarge.4.linux.bigdata",
            root_volume=rootVolumeNodeGroups,
            data_volume=dataVolumeNodeGroups,
            data_volume_count=1,
            assigned_roles=listAssignedRolesNodeGroups
        )
    ]
    listTagsbody = [
        Tag(
            key="tag1",
            value="111"
        ),
        Tag(
            key="tag2",
            value="222"
        )
    ]
    chargeInfobody = ChargeInfo(
        charge_mode="postPaid"
    )
    request.body = RunJobFlowCommand(
        steps=listStepsbody,
        delete_when_no_steps=True,
        log_uri="obs://bucketTest/logs",
        node_groups=listNodeGroupsbody,
        log_collection=1,
        tags=listTagsbody,
        template_id="mgmt_control_combined_v2",
        mrs_ecs_default_agency="MRS_ECS_DEFAULT_AGENCY",
        node_root_password="your password",
        login_mode="PASSWORD",
        manager_admin_password="your password",
        safe_mode="KERBEROS",
        availability_zone="",
        components="Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
```

```
            subnet_name="subnet",
            subnet_id="1f8c5ca6-1f66-4096-bb00-baf175954f6e",
            vpc_name="vpc-37cd",
            region="",
            charge_info=chargeInfobody,
            cluster_type="CUSTOM",
            cluster_name="mrs_heshe_dm",
            cluster_version="MRS 3.1.0"
        )
        response = client.run_job_flow(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建一个版本号为MRS 3.1.0的自定义管控合设集群，并提交一个作业类型为HiveScript的作业。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RunJobFlowRequest{}
    var listArgumentsJobExecution = []string{
        "obs://test/hive/sql/HiveScript.sql",
    }
    jobExecutionSteps := &model.JobExecution{
        JobType: "HiveScript",
        JobName: "hive_test",
        Arguments: &listArgumentsJobExecution,
    }
    var listArgumentsJobExecution1 = []string{
        "obs://test/test.sql",
        "/user/hive/input",
    }
    jobExecutionSteps1 := &model.JobExecution{
        JobType: "DistCp",
```

```
    JobName: "import_file",
    Arguments: &listArgumentsJobExecution1,
}
var listStepsbody = []model.StepConfig{
    {
        JobExecution: jobExecutionSteps1,
    },
    {
        JobExecution: jobExecutionSteps,
    },
}
var listAssignedRolesNodeGroups = []string{
    "NodeManager",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "FlinkResource",
}
dataVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups1 = []string{
    "DataNode",
    "NodeManager",
    "RegionServer",
    "Flume:1",
    "Broker",
    "Supervisor",
    "Logviewer",
    "HBaseIndexer",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSBroker:1,2",
    "ThriftServer",
    "ThriftServer1",
    "RESTServer",
    "FlinkResource",
}
dataVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups1 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
var listAssignedRolesNodeGroups2 = []string{
    "OMSServer:1,2",
    "SlapdServer:1,2",
    "KerberosServer:1,2",
    "KerberosAdmin:1,2",
    "quorumpeer:1,2,3",
    "NameNode:2,3",
    "Zkfc:2,3",
    "JournalNode:1,2,3",
    "ResourceManager:2,3",
    "JobHistoryServer:2,3",
    "DBServer:1,3",
    "Hue:1,3",
    "LoaderServer:1,3",
    "MetaStore:1,2,3",
    "WebHCat:1,2,3",
    "HiveServer:1,2,3",
```

```
    "HMaster:2,3",
    "MonitorServer:1,2",
    "Nimbus:1,2",
    "UI:1,2",
    "JDBCServer2x:1,2,3",
    "JobHistory2x:2,3",
    "SparkResource2x:1,2,3",
    "oozie:2,3",
    "LoadBalancer:2,3",
    "TezUI:1,3",
    "TimelineServer:3",
    "RangerAdmin:1,2",
    "UserSync:2",
    "TagSync:2",
    "KerberosClient",
    "SlapdClient",
    "meta",
    "HSConsole:2,3",
    "FlinkResource:1,2,3",
    "DataNode:1,2,3",
    "NodeManager:1,2,3",
    "IndexServer2x:1,2",
    "ThriftServer:1,2,3",
    "RegionServer:1,2,3",
    "ThriftServer1:1,2,3",
    "RESTServer:1,2,3",
    "Broker:1,2,3",
    "Supervisor:1,2,3",
    "Logviewer:1,2,3",
    "Flume:1,2,3",
    "HSBroker:1,2,3",
}
dataVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(600),
}
rootVolumeNodeGroups2 := &model.Volume{
    Type: "SAS",
    Size: int32(480),
}
dataVolumeCountNodeGroups:= int32(1)
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeCountNodeGroups2:= int32(1)
var listNodeGroupsbody = []model.NodeGroupV2{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(3),
        NodeSize: "Sit3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups2,
        DataVolume: dataVolumeNodeGroups2,
        DataVolumeCount: &dataVolumeCountNodeGroups,
        AssignedRoles: &listAssignedRolesNodeGroups2,
    },
    {
        GroupName: "node_group_1",
        NodeNum: int32(3),
        NodeSize: "Sit3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups1,
        DataVolume: dataVolumeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
        AssignedRoles: &listAssignedRolesNodeGroups1,
    },
    {
        GroupName: "node_group_2",
        NodeNum: int32(1),
        NodeSize: "Sit3.4xlarge.4.linux.bigdata",
        RootVolume: rootVolumeNodeGroups,
        DataVolume: dataVolumeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups2,
```

```
            AssignedRoles: &listAssignedRolesNodeGroups,
        },
    }
    var listTagsbody = []model.Tag{
        {
            Key: "tag1",
            Value: "111",
        },
        {
            Key: "tag2",
            Value: "222",
        },
    }
    chargeInfobody := &model.ChargeInfo{
        ChargeMode: "postPaid",
    }
    deleteWhenNoStepsRunJobFlowCommand:= true
    logUriRunJobFlowCommand:= "obs://bucketTest/logs"
    logCollectionRunJobFlowCommand:= model.GetRunJobFlowCommandLogCollectionEnum().E_1
    templateIdRunJobFlowCommand:= "mgmt_control_combined_v2"
    mrsEcsDefaultAgencyRunJobFlowCommand:= "MRS_ECS_DEFAULT_AGENCY"
    nodeRootPasswordRunJobFlowCommand:= "your password"
    subnetIdRunJobFlowCommand:= "1f8c5ca6-1f66-4096-bb00-baf175954f6e"
    request.Body = &model.RunJobFlowCommand{
        Steps: listStepsbody,
        DeleteWhenNoSteps: &deleteWhenNoStepsRunJobFlowCommand,
        LogUri: &logUriRunJobFlowCommand,
        NodeGroups: listNodeGroupsbody,
        LogCollection: &logCollectionRunJobFlowCommand,
        Tags: &listTagsbody,
        TemplateId: &templateIdRunJobFlowCommand,
        MrsEcsDefaultAgency: &mrsEcsDefaultAgencyRunJobFlowCommand,
        NodeRootPassword: &nodeRootPasswordRunJobFlowCommand,
        LoginMode: "PASSWORD",
        ManagerAdminPassword: "your password",
        SafeMode: "KERBEROS",
        AvailabilityZone: "",
        Components: "Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,Ranger,Tez",
        SubnetName: "subnet",
        SubnetId: &subnetIdRunJobFlowCommand,
        VpcName: "vpc-37cd",
        Region: "",
        ChargeInfo: chargeInfobody,
        ClusterType: "CUSTOM",
        ClusterName: "mrs_heshe_dm",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.RunJobFlow(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	正常响应示例。

错误码

请参见**错误码**。

6.1.4 扩容集群

功能介绍

对MRS集群进行扩容。

调用方法

请参见**如何调用API**。

URI

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

表 6-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-42 请求 Body 参数

参数	是否必选	参数类型	描述
node_group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
count	是	Integer	**参数解释：** 扩容节点数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
skip_bootstrap_scripts	否	Boolean	**参数解释：** 扩容时是否在新增节点上跳过执行创建集群时指定的引导操作。 **约束限制：** 不涉及 **取值范围：** ● true：跳过执行创建集群时指定的引导操作。 ● false：不跳过执行创建集群时指定的引导操作。 **默认取值：** true

参数	是否必选	参数类型	描述
scale_without_start	否	Boolean	**参数解释：** 扩容后是否选择不启动扩容节点上的组件。 **约束限制：** 不涉及 **取值范围：** ● true：扩容后不启动组件。 ● false：扩容后启动组件。 **默认取值：** false

响应参数

状态码： 200

表 6-43 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 请求操作结果。非包周期节点组扩容请求下发成功时，会包含该字段且内容为success。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed为操作失败 **默认取值：** 不涉及

参数	参数类型	描述
order_id	String	**参数解释：** 订单ID。对包周期节点组进行扩容时，会返回本次扩容产生的订单ID，需要客户到订单支付页面进行自主支付才能真正触发扩容。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- 在节点组node_group_1上扩容1个节点， 默认跳过引导操作，默认启动组件。

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/expand

{
  "node_group_name" : "node_group_1",
  "count" : "1"
}
```

- 在节点组node_group_1上扩容1个节点， 不跳过引导操作， 不启动组件。

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/expand

{
  "node_group_name" : "node_group_1",
  "count" : "1",
  "skip_bootstrap_scripts" : false,
  "scale_without_start" : true
}
```

响应示例

状态码： 200

按需和包周期节点组扩容时有不同的返回体，包周期节点组返回订单ID，后续需要根据订单ID进行支付才能真正进行扩容。

- 按需节点组扩容成功。

```
{
  "result" : "succeeded"
}
```

- 包周期节点组扩容成功，待支付订单ID为CS231XXXXXXXXXX。

```
{
  "order_id" : "CS231XXXXXXXXXX"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 在节点组node_group_1上扩容1个节点， 默认跳过引导操作，默认启动组件。

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

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


public class ExpandClusterSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ExpandClusterRequest request = new ExpandClusterRequest();
        request.withClusterId("{cluster_id}");
        ExpandParam body = new ExpandParam();
        body.withCount(1);
        body.withNodeGroupName("node_group_1");
        request.withBody(body);
        try {
            ExpandClusterResponse response = client.expandCluster(request);
            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());
        }
    }
}
```

- 在节点组node_group_1上扩容1个节点， 不跳过引导操作， 不启动组件。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
```



```
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ExpandClusterSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ExpandClusterRequest request = new ExpandClusterRequest();
        request.withClusterId("{cluster_id}");
        ExpandParam body = new ExpandParam();
        body.withScaleWithoutStart(true);
        body.withSkipBootstrapScripts(false);
        body.withCount(1);
        body.withNodeGroupName("node_group_1");
        request.withBody(body);
        try {
            ExpandClusterResponse response = client.expandCluster(request);
            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

- 在节点组node_group_1上扩容1个节点， 默认跳过引导操作，默认启动组件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ExpandClusterRequest()
        request.cluster_id = "{cluster_id}"
        request.body = ExpandParam(
            count=1,
            node_group_name="node_group_1"
        )
        response = client.expand_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)
```

- 在节点组node_group_1上扩容1个节点， 不跳过引导操作， 不启动组件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ExpandClusterRequest()
        request.cluster_id = "{cluster_id}"
        request.body = ExpandParam(
            scale_without_start=True,
            skip_bootstrap_scripts=False,
            count=1,
            node_group_name="node_group_1"
        )
        response = client.expand_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

- 在节点组node_group_1上扩容1个节点， 默认跳过引导操作，默认启动组件。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExpandClusterRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.ExpandParam{
        Count: int32(1),
        NodeGroupName: "node_group_1",
    }
    response, err := client.ExpandCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 在节点组node_group_1上扩容1个节点， 不跳过引导操作， 不启动组件。

```
package main

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

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ExpandClusterRequest{}
    request.ClusterId = "{cluster_id}"
    scaleWithoutStartExpandParam:= true
    skipBootstrapScriptsExpandParam:= false
    request.Body = &model.ExpandParam{
        ScaleWithoutStart: &scaleWithoutStartExpandParam,
        SkipBootstrapScripts: &skipBootstrapScriptsExpandParam,
        Count: int32(1),
        NodeGroupName: "node_group_1",
    }
    response, err := client.ExpandCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	按需和包周期节点组扩容时有不同的返回体，包周期节点组返回订单ID，后续需要根据订单ID进行支付才能真正进行扩容。

错误码

请参见**错误码**。

6.1.5 缩容集群

功能介绍

对MRS集群进行缩容。

调用方法

请参见**如何调用API**。

URI

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

表 6-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-45 请求 Body 参数

参数	是否必选	参数类型	描述
node_group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
count	否	Integer	**参数解释：** 缩容节点数量，如果是指定节点缩容，则该参数可以不填。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_ids	否	Array of strings	**参数解释：** 缩容节点时指定待删除节点的资源ID列表。resource_ids为空时，按照系统规则自动选择删除节点。仅支持删除状态异常的ecs节点。会针对指定节点进行强制删除。可通过查询主机接口获取resource_id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-46 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 更新映射请求操作结果。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed：操作失败 **默认取值：** 不涉及

请求示例

- 在node_group_1节点组上缩容1个节点。

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/shrink

{
  "node_group_name" : "node_group_1",
  "count" : 1
}
```

- 在node_group_1节点组上强制缩容资源id为678050cd-ba1d-4550-942d-f2e396b1c6fb的节点。

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/shrink

{
  "node_group_name" : "node_group_1",
  "resource_ids" : [ "678050cd-ba1d-4550-942d-f2e396b1c6fb" ]
}
```

响应示例

状态码： 200

缩容成功

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 在node_group_1节点组上缩容1个节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ShrinkClusterSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShrinkClusterRequest request = new ShrinkClusterRequest();
        request.withClusterId("{cluster_id}");
        ShrinkParam body = new ShrinkParam();
        body.withCount(1);
        body.withNodeGroupName("node_group_1");
        request.withBody(body);
        try {
            ShrinkClusterResponse response = client.shrinkCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 在node_group_1节点组上强制缩容资源id为678050cd-ba1d-4550-942d-f2e396b1c6fb的节点。

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

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

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

public class ShrinkClusterSolution {
```

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShrinkClusterRequest request = new ShrinkClusterRequest();
        request.withClusterId("{cluster_id}");
        ShrinkParam body = new ShrinkParam();
        List<String> listbodyResourceIds = new ArrayList<>();
        listbodyResourceIds.add("678050cd-ba1d-4550-942d-f2e396b1c6fb");
        body.withResourceIds(listbodyResourceIds);
        body.withNodeGroupName("node_group_1");
        request.withBody(body);
        try {
            ShrinkClusterResponse response = client.shrinkCluster(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 在node_group_1节点组上缩容1个节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

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

try:
    request = ShrinkClusterRequest()
    request.cluster_id = "{cluster_id}"
    request.body = ShrinkParam(
        count=1,
        node_group_name="node_group_1"
    )
    response = client.shrink_cluster(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 在node_group_1节点组上强制缩容资源id为678050cd-ba1d-4550-942d-f2e396b1c6fb的节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShrinkClusterRequest()
        request.cluster_id = "{cluster_id}"
        listResourceIdsbody = [
            "678050cd-ba1d-4550-942d-f2e396b1c6fb"
        ]
        request.body = ShrinkParam(
            resource_ids=listResourceIdsbody,
            node_group_name="node_group_1"
        )
        response = client.shrink_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 在node_group_1节点组上缩容1个节点。

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShrinkClusterRequest{}
    request.ClusterId = "{cluster_id}"
    countShrinkParam:= int32(1)
    request.Body = &model.ShrinkParam{
        Count: &countShrinkParam,
        NodeGroupName: "node_group_1",
    }
    response, err := client.ShrinkCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 在node_group_1节点组上强制缩容资源id为678050cd-ba1d-4550-942d-f2e396b1c6fb的节点。

```
package main

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

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShrinkClusterRequest{}
    request.ClusterId = "{cluster_id}"
    var listResourceIdsbody = List{
        "678050cd-ba1d-4550-942d-f2e396b1c6fb",
    }
    request.Body = &model.ShrinkParam{
        ResourceIds: &listResourceIdsbody,
        NodeGroupName: "node_group_1",
    }
    response, err := client.ShrinkCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	缩容成功

错误码

请参见**错误码**。

6.1.6 集群添加组件

功能介绍

集群添加组件，仅MRS 3.1.2及之后普通版本和MRS 3.1.2-LTS.2及之后的LTS版本的自定义集群支持添加组件功能。

调用方法

请参见**如何调用API**。

URI

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

表 6-47 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-48 请求 Body 参数

参数	是否必选	参数类型	描述
components_install_mode	是	Array of **ComponentInstallMode** objects	**参数解释：** 组件模型详情。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-49 ComponentInstallMode

参数	是否必选	参数类型	描述
component	是	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
node_groups	是	Array of **AssignedNodeGroup** objects	**参数解释：** 该组件的角色部署信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
component_user_password	否	String	**参数解释：** 配置组件用户密码，该密码用于ClickHouse组件机机用户连接使用。 **约束限制：** 不涉及 **取值范围：** ● 密码长度应在8-26个字符之间 ● 不能与用户名或者倒序用户名相同 ● 必须包含如下4种字符的组合 ● 至少一个小写字母 ● 至少一个大写字母 ● 至少一个数字 ● 至少一个特殊字符：!@$%^-_=+[{}]:,./? **默认取值：** 不涉及
component_default_password	否	String	**参数解释：** 配置组件default用户密码，该密码用于ClickHouse组件人机用户连接使用。 **约束限制：** 不涉及 **取值范围：** ● 密码长度应在8-26个字符之间 ● 不能与用户名或者倒序用户名相同 ● 必须包含如下4种字符的组合 ● 至少一个小写字母 ● 至少一个大写字母 ● 至少一个数字 ● 至少一个特殊字符：!@$%^-_=+[{}]:,./? **默认取值：** 不涉及

表 6-50 AssignedNodeGroup

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
assigned_roles	是	Array of strings	**参数解释：** 角色部署信息。可以指定节点组中部署的角色，该参数是一个字符串数组，每个字符串表示一个角色表达式。 角色表达式定义： • 当该角色在节点组所有节点部署时： {role name}，如“DataNode”。 • 当该角色在节点组指定下标节点部署时：{role name}:{index1},{index2}…,{indexN}，如“NameNode:1,2”，下标从1开始计数。 可选的角色请参考**MRS支持的角色与组件对应表**。 **约束限制：** 不能超过1000条。 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-51 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 更新映射请求操作结果。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed：操作失败 **默认取值：** 不涉及

请求示例

- 非安全集群添加请求添加ClickHouse组件

```
v2/f77c10d14a544393a24e5f0bf53202b6/clusters/ff879d3a-e5d5-4485-a9b6-c673b52673fa/
components

{
  "components_install_mode" : [ {
    "component" : "ClickHouse",
    "node_groups" : [ {
      "name" : "master_node_default_group",
      "assigned_roles" : [ "ClickHouseServer:1,2" ]
    }, {
      "name" : "node_group_1",
      "assigned_roles" : [ "ClickHouseServer", "ClickHouseBalancer" ]
    } ],
    "component_user_password" : "*****",
    "component_default_password" : "*****"
  } ]
}
```

- 添加Hbase组件

```
v2/f77c10d14a544393a24e5f0bf53202b6/clusters/ff879d3a-e5d5-4485-a9b6-c673b52673fa/
components

{
  "components_install_mode" : [ {
    "component" : "HBase",
    "node_groups" : [ {
      "name" : "master_node_default_group",
      "assigned_roles" : [ "RegionServer", "HMaster" ]
    } ]
  } ]
}
```

响应示例

状态码： 200

请求处理结果

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 非安全集群添加请求添加ClickHouse组件

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

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

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

public class AddComponentSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        AddComponentRequest request = new AddComponentRequest();
        request.withClusterId("{cluster_id}");
        AddComponentsReq body = new AddComponentsReq();
        List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
        listNodeGroupsAssignedRoles.add("ClickHouseServer");
        listNodeGroupsAssignedRoles.add("ClickHouseBalancer");
        List<String> listNodeGroupsAssignedRoles1 = new ArrayList<>();
        listNodeGroupsAssignedRoles1.add("ClickHouseServer:1,2");
        List<AssignedNodeGroup> listComponentsInstallModeNodeGroups = new ArrayList<>();
        listComponentsInstallModeNodeGroups.add(
            new AssignedNodeGroup()
                .withName("master_node_default_group")
                .withAssignedRoles(listNodeGroupsAssignedRoles1)
        );
        listComponentsInstallModeNodeGroups.add(
            new AssignedNodeGroup()
                .withName("node_group_1")
                .withAssignedRoles(listNodeGroupsAssignedRoles)
        );
        List<ComponentInstallMode> listbodyComponentsInstallMode = new ArrayList<>();
        listbodyComponentsInstallMode.add(
            new ComponentInstallMode()
                .withComponent("ClickHouse")
                .withNodeGroups(listComponentsInstallModeNodeGroups)
                .withComponentUserPassword("*****")
```

```
            .withComponentDefaultPassword("*****")
        );
        body.withComponentsInstallMode(listbodyComponentsInstallMode);
        request.withBody(body);
        try {
            AddComponentResponse response = client.addComponent(request);
            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());
        }
    }
}
```

- 添加Hbase组件

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

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

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

public class AddComponentSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        AddComponentRequest request = new AddComponentRequest();
        request.withClusterId("{cluster_id}");
        AddComponentsReq body = new AddComponentsReq();
        List<String> listNodeGroupsAssignedRoles = new ArrayList<>();
        listNodeGroupsAssignedRoles.add("RegionServer");
        listNodeGroupsAssignedRoles.add("HMaster");
        List<AssignedNodeGroup> listComponentsInstallModeNodeGroups = new ArrayList<>();
        listComponentsInstallModeNodeGroups.add(
            new AssignedNodeGroup()
                .withName("master_node_default_group")
                .withAssignedRoles(listNodeGroupsAssignedRoles)
        );
```

```
        List<ComponentInstallMode> listbodyComponentsInstallMode = new ArrayList<>();
        listbodyComponentsInstallMode.add(
            new ComponentInstallMode()
                .withComponent("HBase")
                .withNodeGroups(listComponentsInstallModeNodeGroups)
        );
        body.withComponentsInstallMode(listbodyComponentsInstallMode);
        request.withBody(body);
        try {
            AddComponentResponse response = client.addComponent(request);
            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

- 非安全集群添加请求添加ClickHouse组件

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddComponentRequest()
        request.cluster_id = "{cluster_id}"
        listAssignedRolesNodeGroups = [
            "ClickHouseServer",
            "ClickHouseBalancer"
        ]
        listAssignedRolesNodeGroups1 = [
            "ClickHouseServer:1,2"
        ]
        listNodeGroupsComponentsInstallMode = [
            AssignedNodeGroup(
                name="master_node_default_group",
                assigned_roles=listAssignedRolesNodeGroups1
            ),
```

```
        AssignedNodeGroup(
            name="node_group_1",
            assigned_roles=listAssignedRolesNodeGroups
        )
    ]
    listComponentsInstallModebody = [
        ComponentInstallMode(
            component="ClickHouse",
            node_groups=listNodeGroupsComponentsInstallMode,
            component_user_password="*****",
            component_default_password="*****"
        )
    ]
    request.body = AddComponentsReq(
        components_install_mode=listComponentsInstallModebody
    )
    response = client.add_component(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 添加Hbase组件

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddComponentRequest()
        request.cluster_id = "{cluster_id}"
        listAssignedRolesNodeGroups = [
            "RegionServer",
            "HMaster"
        ]
        listNodeGroupsComponentsInstallMode = [
            AssignedNodeGroup(
                name="master_node_default_group",
                assigned_roles=listAssignedRolesNodeGroups
            )
        ]
        listComponentsInstallModebody = [
            ComponentInstallMode(
                component="HBase",
                node_groups=listNodeGroupsComponentsInstallMode
            )
        ]
```

```
        request.body = AddComponentsReq(
            components_install_mode=listComponentsInstallModebody
        )
        response = client.add_component(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 非安全集群添加请求添加ClickHouse组件

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddComponentRequest{}
    request.ClusterId = "{cluster_id}"
    var listAssignedRolesNodeGroups = []string{
        "ClickHouseServer",
        "ClickHouseBalancer",
    }
    var listAssignedRolesNodeGroups1 = []string{
        "ClickHouseServer:1,2",
    }
    var listNodeGroupsComponentsInstallMode = []model.AssignedNodeGroup{
        {
            Name: "master_node_default_group",
            AssignedRoles: listAssignedRolesNodeGroups1,
        },
        {
            Name: "node_group_1",
            AssignedRoles: listAssignedRolesNodeGroups,
        },
    }
    componentUserPasswordComponentsInstallMode:= "*****"
    componentDefaultPasswordComponentsInstallMode:= "*****"
    var listComponentsInstallModebody = []model.ComponentInstallMode{
```

```
        {
            Component: "ClickHouse",
            NodeGroups: listNodeGroupsComponentsInstallMode,
            ComponentUserPassword: &componentUserPasswordComponentsInstallMode,
            ComponentDefaultPassword: &componentDefaultPasswordComponentsInstallMode,
        },
    }
    request.Body = &model.AddComponentsReq{
        ComponentsInstallMode: listComponentsInstallModebody,
    }
    response, err := client.AddComponent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 添加Hbase组件

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddComponentRequest{}
    request.ClusterId = "{cluster_id}"
    var listAssignedRolesNodeGroups = []string{
        "RegionServer",
        "HMaster",
    }
    var listNodeGroupsComponentsInstallMode = []model.AssignedNodeGroup{
        {
            Name: "master_node_default_group",
            AssignedRoles: listAssignedRolesNodeGroups,
        },
    }
    var listComponentsInstallModebody = []model.ComponentInstallMode{
        {
            Component: "HBase",
            NodeGroups: listNodeGroupsComponentsInstallMode,
        },
    }
```

```
    request.Body = &model.AddComponentsReq{
        ComponentsInstallMode: listComponentsInstallModebody,
    }
    response, err := client.AddComponent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	请求处理结果

错误码

请参见**错误码**。

6.1.7 查询集群节点列表

功能介绍

查询集群节点列表。

调用方法

请参见**如何调用API**。

URI

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

表 6-52 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 6-53 Query 参数

参数	是否必选	参数类型	描述
node_group	否	String	**参数解释：** 要查询的节点组名称。 **约束限制：** 不涉及 **取值范围：** 长度为[3-14]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
limit	否	Integer	**参数解释：** 返回结果中每页显示条数。 **约束限制：** 不涉及 **取值范围：** ≥1 **默认取值：** 10
offset	否	Integer	**参数解释：** 表示作业列表从该偏移量开始查询。 **约束限制：** 不涉及 **取值范围：** ≥1 **默认取值：** 1
node_name	否	String	**参数解释：** 指定节点名称，支持模糊搜索。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
sort_key	否	String	**参数解释：** 排序键，支持对节点名称排序。 **约束限制：** 不涉及 **取值范围：** node_name：节点名称。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
sort_dir	否	String	**参数解释：** 列表排序方式，desc为降序，asc为升序。 **约束限制：** 不涉及 **取值范围：** desc：降序排列 asc：升序排列 **默认取值：** 不涉及
query_node_detail	否	Boolean	**参数解释：** 是否查询节点详情。该字段设为true时可能会影响接口性能。 **约束限制：** 不涉及 **取值范围：** true：查询节点详情 false：不查询节点详情 **默认取值：** false
query_ecs_detail	否	Boolean	**参数解释：** 是否查询ECS详情信息，会涉及对ECS接口调用。 **约束限制：** 不涉及 **取值范围：** true：查询ECS详情信息 false：不查询ECS详情信息 **默认取值：** false
internal_ip	否	String	**参数解释：** 指定内网IP。 **约束限制：** 不涉及 **取值范围：** 长度为[7-15]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-54 响应 Body 参数

参数	参数类型	描述
nodes	Array of **ClusterNode** objects	**参数解释：** 节点列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_total	Integer	**参数解释：** 节点数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-55 ClusterNode

参数	参数类型	描述
node_name	String	**参数解释：** 节点名称，对应manager里的节点名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_id	String	**参数解释：** 资源id。确定节点的唯一性，包周期节点可用于计费的查询。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_group_name	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_type	String	**参数解释：** 节点类型。Task、Core、Master等。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
billing_type	String	**参数解释：** on-period包周期或者on-quantity按需。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
deployment_type	String	**参数解释：** 部署类型。支持Server主机类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
server_info	**ServerInfo** object	**参数解释：** 如果部署类型为Server类型，则该字段不为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	Array of **Tag** objects	**参数解释：** 节点标签。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
node_detail	**NodeDetail** object	**参数解释：** 节点监控信息。需要IAM同步后才能返回。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_status	String	**参数解释：** 节点状态。对应页面上的操作状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
component_infos	Array of **ComponentInfo** objects	**参数解释：** 组件实例信息数组。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-56 ServerInfo

参数	参数类型	描述
server_id	String	**参数解释：** 服务器ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
server_name	String	**参数解释：** 服务器名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
server_type	String	**参数解释：** 服务器类型。ECS或者BMS。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
data_volumes	Array of **VolumeInfo** objects	**参数解释：** 数据盘。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume	**VolumeInfo** object	**参数解释：** 系统盘。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
cpu_type	String	**参数解释：** CPU类型。x86或者ARM。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cpu	String	**参数解释：** CPU大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
mem	String	**参数解释：** 内存大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
internal_ip	String	**参数解释：** 内部IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-57 VolumeInfo

参数	参数类型	描述
type	String	**参数解释：** 磁盘类型。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO磁盘类型。 ● SAS：高IO磁盘类型。 ● SSD：超高IO磁盘类型。 ● GPSSD：通用型SSD磁盘类型。 **默认取值：** 不涉及
size	Integer	**参数解释：** 磁盘大小。单位为GB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
count	Integer	**参数解释：** 磁盘数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-58 Tag

参数	参数类型	描述
key	String	**参数解释：** 标签键。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
value	String	**参数解释：** 标签值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-59 NodeDetail

参数	参数类型	描述
running_status	String	**参数解释：** 运行状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cpu_usage	String	**参数解释：** CPU使用率。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
memory_usage	String	**参数解释：** 内存使用率。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
disk_usage	String	**参数解释：** 硬盘使用率。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
total_memory	String	**参数解释：** 总内存。单位MB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
available_memory	String	**参数解释：** 可用内存。单位MB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
total_hard_disk_space	String	**参数解释：** 总硬盘空间。单位GB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
available_hard_disk_space	String	**参数解释：** 可用硬盘空间。单位GB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
network_read	String	**参数解释：** 网络读取速度。单位Byte/s。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
network_write	String	**参数解释：** 网络写入速度。单位Byte/s。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-60 ComponentInfo

参数	参数类型	描述
id	String	**参数解释:** 组件ID。 **约束限制:** 不涉及 **取值范围:** 不涉及 **默认取值:** 不涉及
name	String	**参数解释:** 组件名。 **约束限制:** 不涉及 **取值范围:** 不涉及 **默认取值:** 不涉及
instance_group_name	String	**参数解释:** 组件所在组名称。 **约束限制:** 不涉及 **取值范围:** 不涉及 **默认取值:** 不涉及
running_status	String	**参数解释:** 运行状态。 **约束限制:** 不涉及 **取值范围:** 不涉及 **默认取值:** 不涉及

参数	参数类型	描述
ha_status	String	**参数解释：** HA状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
config_status	String	**参数解释：** 配置状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
role_name	String	**参数解释：** 角色名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
role_short_name	String	**参数解释：** 角色缩写。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
role_type	String	**参数解释：** 角色类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
service_name	String	**参数解释：** 服务名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
pair_name	String	**参数解释：** 对名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
relation_pairs	String	**参数解释：** 关联对。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
support_decom	String	**参数解释：** 是否支持Decom。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
support_reinstall	String	**参数解释：** 是否支持重装。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
support_collect_stack_info	String	**参数解释：** 是否支持收集堆栈信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询集群指定节点组的节点列表。

```
https://mrs.cn-north-7.myhuaweicloud.com/v2/174ee662a7e24cc99bfc858c4558dbf6/clusters/
f0a91b14-8884-4ba1-9e8d-0a21086c0ab4/nodes?node_group=master_node_default_group
```

响应示例

状态码： 200

查询的节点列表信息。

```
{
  "nodes" : [ {
    "node_name" : "node-master1QxCW",
    "resource_id" : "fc1ed6bb-e3d8-4dc8-8162-ef673bff6b7b",
    "node_group_name" : "master_node_default_group",
    "node_type" : "Master",
```

```
  "billing_type" : "on-quantity",
  "deployment_type" : "SERVER",
  "server_info" : {
    "server_id" : "3687f50a-cd4e-4c67-8858-5d6555c8834c",
    "server_name" : "f28fb043-ecbc-401e-936e-fb321ca8d40a_node_master1QxCW",
    "server_type" : "ECS",
    "data_volumes" : [ {
      "type" : "SATA",
      "size" : 200,
      "count" : 1
    } ],
    "root_volume" : {
      "type" : "SATA",
      "size" : 100,
      "count" : 1
    },
    "cpu_type" : "X86",
    "internal_ip" : "192.168.10.142"
  },
  "tags" : [ {
    "key" : "1",
    "value" : "2"
  } ],
  "node_status" : "started"
 } ],
 "node_total" : 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.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ListNodesSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListNodesRequest request = new ListNodesRequest();
```

```
        request.withClusterId("{cluster_id}");
        try {
            ListNodesResponse response = client.listNodes(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListNodesRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_nodes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

func main() {
```

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	查询的节点列表信息。

错误码

请参见**错误码**。

6.2 作业管理接口

6.2.1 新增并执行作业

功能介绍

在MRS集群中新增并提交一个作业。

需要先在集群详情页的“概览”页签，单击“IAM用户同步”右侧的“同步”进行IAM用户同步，然后再通过该接口提交作业。

如需使用OBS加密功能，请先参考“MRS用户指南 > 管理现有集群 > 作业管理 > 使用OBS加密数据运行作业”页面进行相关配置后，再调用API接口运行作业。

所有示例中涉及的OBS路径、样例文件及终端节点和AKSK，请提前准备并在提交请求时根据实际情况替换。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/job-executions

表 6-61 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-62 请求 Body 参数

参数	是否必选	参数类型	描述
job_type	是	String	**参数解释：** 作业类型。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● SparkSubmit ● SparkPython：该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。 ● HiveScript ● HiveSql ● DistCp，导入、导出数据。 ● SparkScript ● SparkSql ● Flink **默认取值：** 不涉及
job_name	是	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
arguments	否	Array of strings	**参数解释：** 程序执行的关键参数，该参数由用户程序内的函数指定，MRS只负责参数的传入。 **约束限制：** 参数最多为150000字符，不能包含;|&>'<$!'"\特殊字符，可为空。 说明： • 若输入带有敏感信息（如登录密码）的参数可能在作业详情展示和日志打印中存在暴露的风险，请谨慎操作。 • 提交HiveScript或HiveSql类型的作业时如需以“obs://”开头格式访问存储在OBS上的文件，请在Hive服务配置页面搜索参数“core.site.customized.configs”，新增OBS的endpoint配置项，参数为“fs.obs.endpoint”，值请输入OBS对应的endpoint，具体请参考**终端节点**。 **取值范围：** 不涉及 **默认取值：** 不涉及
properties	否	Map<String,String>	**参数解释：** 程序系统参数。 **约束限制：** 参数最多为2048字符，不能包含><|'`&!\特殊字符，可为空。 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 202

表 6-63 响应 Body 参数

参数	参数类型	描述
job_submit_result	**JobSubmitResult** object	**参数解释：** 作业执行结果。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-64 JobSubmitResult

参数	参数类型	描述
job_id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
state	String	**参数解释：** 作业提交状态。 **约束限制：** 不涉及 **取值范围：** ● COMPLETE：作业提交完成。 ● FAILED：作业提交失败。 **默认取值：** 不涉及

状态码： 400

表 6-65 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- 新增一个MapReduce作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "MapReduceTest",
  "job_type" : "MapReduce",
  "arguments" : [ "obs://obs-test/program/hadoop-mapreduce-examples-x.x.x.jar", "wordcount", "obs://obs-test/input/", "obs://obs-test/job/mapreduce/output" ],
  "properties" : {
    "fs.obs.endpoint" : "obs endpoint",
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新增一个SparkSubmit作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "SparkSubmitTest",
  "job_type" : "SparkSubmit",
  "arguments" : [ "--master", "yarn", "--deploy-mode", "cluster", "--py-files", "obs://obs-test/a.py", "--conf", "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH", "--conf", "spark.yarn.appMasterEnv.aaa=aaaa", "--conf", "spark.executorEnv.aaa=executoraaa", "--properties-file", "obs://obs-test/test-spark.conf", "obs://obs-test/pi.py", "100000" ],
  "properties" : {
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新增一个HiveScript作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "HiveScriptTest",
  "job_type" : "HiveScript",
  "arguments" : [ "obs://obs-test/sql/test_script.sql" ],
  "properties" : {
    "fs.obs.endpoint" : "obs endpoint",
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新建一个HiveSql作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "HiveSqlTest",
  "job_type" : "HiveSql",
  "arguments" : [ "DROP TABLE IF EXISTS src_wordcount;\ncreate external table src_wordcount(line string) row format delimited fields terminated by \"\\n\" stored as textfile location \"obs://donotdel-gxc/input/\";\ninsert into src_wordcount values(\"v1\")" ],
  "properties" : {
    "fs.obs.endpoint" : "obs endpoint",
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新建一个DistCp作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "DistCpTest",
  "job_type" : "DistCp",
  "arguments" : [ "obs://obs-test/DistcpJob/", "/user/test/sparksql/" ],
  "properties" : {
    "fs.obs.endpoint" : "obs endpoint",
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新建一个SparkScript作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_type" : "SparkSql",
  "job_name" : "SparkScriptTest",
  "arguments" : [ "op-key1", "op-value1", "op-key2", "op-value2", "obs://obs-test/sql/test_script.sql" ],
  "properties" : {
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新建一个SparkSql作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_type" : "SparkSql",
  "job_name" : "SparkSqlTest",
  "arguments" : [ "op-key1", "op-value1", "op-key2", "op-value2", "create table student_info3 (id string,name string,gender string,age int,addr string);" ],
  "properties" : {
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
```

```
  }
}
```

- 新建一个Flink作业

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_name" : "flinkTest",
  "job_type" : "Flink",
  "arguments" : [ "run", "-d", "-ynm", "testExcutorejobhdfsbatch", "-m", "yarn-cluster", "hdfs://test/
examples/batch/WordCount.jar" ],
  "properties" : {
    "fs.obs.endpoint" : "obs endpoint",
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

- 新增一个SparkPython作业（该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。）

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions

{
  "job_type" : "SparkPython",
  "job_name" : "SparkPythonTest",
  "arguments" : [ "--master", "yarn", "--deploy-mode", "cluster", "--py-files", "obs://obs-test/a.py", "--
conf", "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH", "--conf",
"spark.yarn.appMasterEnv.aaa=aaaa", "--conf", "spark.executorEnv.aaa=executoraaa", "--properties-
file", "obs://obs-test/test-spark.conf", "obs://obs-test/pi.py", 100000 ],
  "properties" : {
    "fs.obs.access.key" : "xxx",
    "fs.obs.secret.key" : "yyy"
  }
}
```

响应示例

状态码： 202

新增并执行作业

```
{
  "job_submit_result" : {
    "job_id" : "44b37a20-ffe8-42b1-b42b-78a5978d7e40",
    "state" : "COMPLETE"
  }
}
```

状态码： 400

新增并执行作业失败

```
{
  "job_submit_result" : {
    "error_msg" : "不能提交Hive相关作业",
    "error_code" : "0168"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 新增一个MapReduce作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.endpoint", "obs endpoint");
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("obs://obs-test/program/hadoop-mapreduce-examples-x.x.x.jar");
        listbodyArguments.add("wordcount");
        listbodyArguments.add("obs://obs-test/input/");
        listbodyArguments.add("obs://obs-test/job/mapreduce/output");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("MapReduceTest");
        body.withJobType("MapReduce");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新增一个SparkSubmit作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("--master");
        listbodyArguments.add("yarn");
        listbodyArguments.add("--deploy-mode");
        listbodyArguments.add("cluster");
        listbodyArguments.add("--py-files");
        listbodyArguments.add("obs://obs-test/a.py");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.yarn.appMasterEnv.aaa=aaaa");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.executorEnv.aaa=executoraaa");
        listbodyArguments.add("--properties-file");
        listbodyArguments.add("obs://obs-test/test-spark.conf");
        listbodyArguments.add("obs://obs-test/pi.py");
        listbodyArguments.add("100000");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
```

```
        body.withJobName("SparkSubmitTest");
        body.withJobType("SparkSubmit");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新增一个HiveScript作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.endpoint", "obs endpoint");
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("obs://obs-test/sql/test_script.sql");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
```

```
        body.withJobName("HiveScriptTest");
        body.withJobType("HiveScript");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新建一个HiveSql作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.endpoint", "obs endpoint");
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("DROP TABLE IF EXISTS src_wordcount;
create external table src_wordcount(line string) row format delimited fields terminated by "\n" stored as textfile location "obs://donotdel-gxc/input/";
```

```
insert into src_wordcount values("v1")");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("HiveSqlTest");
        body.withJobType("HiveSql");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新建一个DistCp作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.endpoint", "obs endpoint");
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
```

```
        listbodyArguments.add("obs://obs-test/DistcpJob/");
        listbodyArguments.add("/user/test/sparksql/");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("DistCpTest");
        body.withJobType("DistCp");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新建一个SparkScript作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
```

```
        listbodyArguments.add("op-key1");
        listbodyArguments.add("op-value1");
        listbodyArguments.add("op-key2");
        listbodyArguments.add("op-value2");
        listbodyArguments.add("obs://obs-test/sql/test_script.sql");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("SparkScriptTest");
        body.withJobType("SparkSql");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新建一个SparkSql作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
```

```
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("op-key1");
        listbodyArguments.add("op-value1");
        listbodyArguments.add("op-key2");
        listbodyArguments.add("op-value2");
        listbodyArguments.add("create table student_info3 (id string,name string,gender string,age
int,addr string);");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("SparkSqlTest");
        body.withJobType("SparkSql");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新建一个Flink作业

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
```

```
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.endpoint", "obs endpoint");
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("run");
        listbodyArguments.add("-d");
        listbodyArguments.add("-ynm");
        listbodyArguments.add("testExcutorejobhdfsbatch");
        listbodyArguments.add("-m");
        listbodyArguments.add("yarn-cluster");
        listbodyArguments.add("hdfs://test/examples/batch/WordCount.jar");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("flinkTest");
        body.withJobType("Flink");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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());
        }
    }
}
```

- 新增一个SparkPython作业（该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。）

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

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

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateExecuteJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateExecuteJobRequest request = new CreateExecuteJobRequest();
        request.withClusterId("{cluster_id}");
        JobExecution body = new JobExecution();
        Map<String, String> listbodyProperties = new HashMap<>();
        listbodyProperties.put("fs.obs.access.key", "xxx");
        listbodyProperties.put("fs.obs.secret.key", "yyy");
        List<String> listbodyArguments = new ArrayList<>();
        listbodyArguments.add("--master");
        listbodyArguments.add("yarn");
        listbodyArguments.add("--deploy-mode");
        listbodyArguments.add("cluster");
        listbodyArguments.add("--py-files");
        listbodyArguments.add("obs://obs-test/a.py");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.yarn.appMasterEnv.aaa=aaaa");
        listbodyArguments.add("--conf");
        listbodyArguments.add("spark.executorEnv.aaa=executoraaa");
        listbodyArguments.add("--properties-file");
        listbodyArguments.add("obs://obs-test/test-spark.conf");
        listbodyArguments.add("obs://obs-test/pi.py");
        listbodyArguments.add("100000");
        body.withProperties(listbodyProperties);
        body.withArguments(listbodyArguments);
        body.withJobName("SparkPythonTest");
        body.withJobType("SparkPython");
        request.withBody(body);
        try {
            CreateExecuteJobResponse response = client.createExecuteJob(request);
            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

- 新增一个MapReduce作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.endpoint": "obs endpoint",
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "obs://obs-test/program/hadoop-mapreduce-examples-x.x.x.jar",
            "wordcount",
            "obs://obs-test/input/",
            "obs://obs-test/job/mapreduce/output"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="MapReduceTest",
            job_type="MapReduce"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新增一个SparkSubmit作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
```

```
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "--master",
            "yarn",
            "--deploy-mode",
            "cluster",
            "--py-files",
            "obs://obs-test/a.py",
            "--conf",
            "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH",
            "--conf",
            "spark.yarn.appMasterEnv.aaa=aaaa",
            "--conf",
            "spark.executorEnv.aaa=executoraaa",
            "--properties-file",
            "obs://obs-test/test-spark.conf",
            "obs://obs-test/pi.py",
            "100000"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="SparkSubmitTest",
            job_type="SparkSubmit"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新增一个HiveScript作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.endpoint": "obs endpoint",
```

```
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "obs://obs-test/sql/test_script.sql"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="HiveScriptTest",
            job_type="HiveScript"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新建一个HiveSql作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.endpoint": "obs endpoint",
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "DROP TABLE IF EXISTS src_wordcount;
        create external table src_wordcount(line string) row format delimited fields terminated by "\n" stored as textfile location "obs://donotdel-gxc/input/";
        insert into src_wordcount values("v1")"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="HiveSqlTest",
            job_type="HiveSql"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新建一个DistCp作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.endpoint": "obs endpoint",
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "obs://obs-test/DistcpJob/",
            "/user/test/sparksql/"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="DistCpTest",
            job_type="DistCp"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新建一个SparkScript作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "op-key1",
            "op-value1",
            "op-key2",
            "op-value2",
            "obs://obs-test/sql/test_script.sql"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="SparkScriptTest",
            job_type="SparkSql"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新建一个SparkSql作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

```
    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "op-key1",
            "op-value1",
            "op-key2",
            "op-value2",
            "create table student_info3 (id string,name string,gender string,age int,addr string);"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="SparkSqlTest",
            job_type="SparkSql"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新建一个Flink作业

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.endpoint": "obs endpoint",
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "run",
            "-d",
            "-ynm",
            "testExcutorejobhdfsbatch",
            "-m",
            "yarn-cluster",
```

```
            "hdfs://test/examples/batch/WordCount.jar"
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="flinkTest",
            job_type="Flink"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 新增一个SparkPython作业（该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。）

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateExecuteJobRequest()
        request.cluster_id = "{cluster_id}"
        listPropertiesbody = {
            "fs.obs.access.key": "xxx",
            "fs.obs.secret.key": "yyy"
        }
        listArgumentsbody = [
            "--master",
            "yarn",
            "--deploy-mode",
            "cluster",
            "--py-files",
            "obs://obs-test/a.py",
            "--conf",
            "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH",
            "--conf",
            "spark.yarn.appMasterEnv.aaa=aaaa",
            "--conf",
            "spark.executorEnv.aaa=executoraaa",
            "--properties-file",
            "obs://obs-test/test-spark.conf",
            "obs://obs-test/pi.py",
            "100000"
```

```
        ]
        request.body = JobExecution(
            properties=listPropertiesbody,
            arguments=listArgumentsbody,
            job_name="SparkPythonTest",
            job_type="SparkPython"
        )
        response = client.create_execute_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 新增一个MapReduce作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.endpoint": "obs endpoint",
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "obs://obs-test/program/hadoop-mapreduce-examples-x.x.x.jar",
        "wordcount",
        "obs://obs-test/input/",
        "obs://obs-test/job/mapreduce/output",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "MapReduceTest",
        JobType: "MapReduce",
```

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

- 新增一个SparkSubmit作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "--master",
        "yarn",
        "--deploy-mode",
        "cluster",
        "--py-files",
        "obs://obs-test/a.py",
        "--conf",
        "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH",
        "--conf",
        "spark.yarn.appMasterEnv.aaa=aaaa",
        "--conf",
        "spark.executorEnv.aaa=executoraaa",
        "--properties-file",
        "obs://obs-test/test-spark.conf",
        "obs://obs-test/pi.py",
        "100000",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
```

```
        JobName: "SparkSubmitTest",
        JobType: "SparkSubmit",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新增一个HiveScript作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.endpoint": "obs endpoint",
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "obs://obs-test/sql/test_script.sql",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "HiveScriptTest",
        JobType: "HiveScript",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新建一个HiveSql作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.endpoint": "obs endpoint",
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "DROP TABLE IF EXISTS src_wordcount;
    create external table src_wordcount(line string) row format delimited fields terminated by "\n" stored as textfile location "obs://donotdel-gxc/input/";
    insert into src_wordcount values("v1")",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "HiveSqlTest",
        JobType: "HiveSql",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新建一个DistCp作业

```
package main

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



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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.endpoint": "obs endpoint",
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "obs://obs-test/DistcpJob/",
        "/user/test/sparksql/",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "DistCpTest",
        JobType: "DistCp",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新建一个SparkScript作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "op-key1",
        "op-value1",
        "op-key2",
        "op-value2",
        "obs://obs-test/sql/test_script.sql",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "SparkScriptTest",
        JobType: "SparkSql",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新建一个SparkSql作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
```

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

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "op-key1",
        "op-value1",
        "op-key2",
        "op-value2",
        "create table student_info3 (id string,name string,gender string,age int,addr string);",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "SparkSqlTest",
        JobType: "SparkSql",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新建一个Flink作业

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.endpoint": "obs endpoint",
        "fs.obs.access.key": "xxx",
```

```
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "run",
        "-d",
        "-ynm",
        "testExcutorejobhdfsbatch",
        "-m",
        "yarn-cluster",
        "hdfs://test/examples/batch/WordCount.jar",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "flinkTest",
        JobType: "Flink",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 新增一个SparkPython作业（该类型作业将转换为SparkSubmit类型提交，MRS控制台界面的作业类型展示为SparkSubmit，通过接口查询作业列表信息时作业类型请选择SparkSubmit。）

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateExecuteJobRequest{}
    request.ClusterId = "{cluster_id}"
    var listPropertiesbody = map[string]string{
        "fs.obs.access.key": "xxx",
        "fs.obs.secret.key": "yyy",
    }
    var listArgumentsbody = []string{
        "--master",
```

```
        "yarn",
        "--deploy-mode",
        "cluster",
        "--py-files",
        "obs://obs-test/a.py",
        "--conf",
        "spark.yarn.appMasterEnv.PYTHONPATH=/tmp:$PYTHONPATH",
        "--conf",
        "spark.yarn.appMasterEnv.aaa=aaaa",
        "--conf",
        "spark.executorEnv.aaa=executoraaa",
        "--properties-file",
        "obs://obs-test/test-spark.conf",
        "obs://obs-test/pi.py",
        "100000",
    }
    request.Body = &model.JobExecution{
        Properties: listPropertiesbody,
        Arguments: &listArgumentsbody,
        JobName: "SparkPythonTest",
        JobType: "SparkPython",
    }
    response, err := client.CreateExecuteJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	新增并执行作业
400	新增并执行作业失败

错误码

请参见**错误码**。

6.2.2 查询作业列表信息

功能介绍

在MRS指定集群中查询作业管理里提交的作业列表信息。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/job-executions

表 6-66 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 6-67 Query 参数

参数	是否必选	参数类型	描述
job_name	否	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-128]个字符。 **默认取值：** 不涉及
job_id	否	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
user	否	String	**参数解释：** 用户名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“-”、“_”和“.”组成，且不能以数字开头，且长度为[1-32]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_type	否	String	**参数解释：** 作业类型。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● SparkPython ● SparkSubmit：SparkPython类型的作业在查询时作业类型请选择SparkSubmit。 ● HiveScript ● HiveSql ● DistCp，导入、导出数据。 ● SparkScript ● SparkSql ● Flink **默认取值：** 不涉及
job_state	否	String	**参数解释：** 作业运行状态。 **约束限制：** 不涉及 **取值范围：** ● FAILED：失败 ● KILLED：已终止 ● NEW：已创建 ● NEW_SAVING：已创建保存中 ● SUBMITTED：已提交 ● ACCEPTED：已接受 ● RUNNING：运行中 ● FINISHED：已完成 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_result	否	String	**参数解释：** 作业运行结果。 **约束限制：** 不涉及 **取值范围：** ● FAILED：执行失败的作业。 ● KILLED：执行中被手动终止的作业。 ● UNDEFINED：正在执行的作业。 ● SUCCEEDED：执行成功的作业。 **默认取值：** 不涉及
queue	否	String	**参数解释：** 作业的资源队列类型名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“-”和“_”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
limit	否	String	**参数解释：** 返回结果中每页显示条数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 10

参数	是否必选	参数类型	描述
offset	否	String	**参数解释：** 表示作业列表从该偏移量开始查询。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 1
sort_by	否	String	**参数解释：** 返回结果的排序方式。 **约束限制：** 不涉及 **取值范围：** ● asc：按升序排列 ● desc：按降序排列 **默认取值：** desc
submitted_time_begin	否	Long	**参数解释：** 查询该时间之后提交的作业，UTC的毫秒时间戳。例如：1562032041362。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
submitted_time_end	否	Long	**参数解释：** 查询该时间之前提交的作业UTC的毫秒时间戳。例如：1562032041362。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 202

表 6-68 响应 Body 参数

参数	参数类型	描述
total_record	Integer	**参数解释：** 总记录数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_list	Array of **JobQueryBean** objects	**参数解释：** 作业列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-69 JobQueryBean

参数	参数类型	描述
job_id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
user	String	**参数解释：** 提交作业的用户名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_name	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_result	String	**参数解释：** 作业最终结果。 **约束限制：** 不涉及 **取值范围：** ● FAILED：执行失败的作业 ● KILLED：执行中被手动终止的作业。 ● UNDEFINED：正在执行的作业。 ● SUCCEEDED：执行成功的作业。 **默认取值：** 不涉及

参数	参数类型	描述
job_state	String	**参数解释：** 作业执行状态。 **约束限制：** 不涉及 **取值范围：** ● FAILED：失败 ● KILLED：已终止 ● NEW：已创建 ● NEW_SAVING：已创建保存中 ● SUBMITTED：已提交 ● ACCEPTED：已接受 ● RUNNING：运行中 ● FINISHED：已完成 **默认取值：** 不涉及
job_progress	Float	**参数解释：** 作业执行进度。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_type	String	**参数解释：** 作业类型。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● SparkSubmit：SparkPython类型的作业在查询时作业类型请选择SparkSubmit。 ● HiveScript ● HiveSql ● DistCp，导入、导出数据。 ● SparkScript ● SparkSql ● Flink **默认取值：** 不涉及
started_time	Long	**参数解释：** 作业开始执行时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
submitted_time	Long	**参数解释：** 作业提交时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
finished_time	Long	**参数解释：** 作业完成时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
elapsed_time	Long	**参数解释：** 作业执行时长。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
arguments	String	**参数解释：** 运行参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
launcher_id	String	**参数解释：** 真实作业编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
properties	String	**参数解释：** 配置参数，用于传-d参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
app_id	String	**参数解释：** 实际作业编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tracking_url	String	**参数解释：** 日志链接地址。当前仅SparkSubmit作业支持该参数。该参数基于集群的EIP访问集群中的YARN WebUI页面，用户如果在VPC界面解绑EIP，MRS服务侧数据会因为未更新导致该参数引用旧EIP导致访问失败，可通过对集群重新进行EIP的绑定来修复该问题。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
queue	String	**参数解释：** 作业的资源队列类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-70 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询作业列表信息请求示例

```
GET https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions
```

响应示例

状态码： 202

查询作业列表信息成功

```
{
  "total_record" : 2,
  "job_list" : [ {
    "job_id" : "981374c1-85da-44ee-be32-edfb4fba776c",
    "user" : "xxxx",
    "job_name" : "SparkSubmitTset",
    "job_result" : "UNDEFINED",
    "job_state" : "ACCEPTED",
    "job_progress" : 0,
    "job_type" : "SparkSubmit",
    "started_time" : 0,
    "submitted_time" : 1564714763119,
    "finished_time" : 0,
    "elapsed_time" : 0,
    "queue" : "default",
    "arguments" : "[--class, --driver-memory, --executor-cores, --master, yarn-cluster, s3a://obs-test/hadoop-mapreduce-examples-3.1.1.jar, dddd]",
```

```
    "launcher_id" : "application_1564622673393_0613",
    "properties" : { }
  }, {
    "job_id" : "c54c8aa0-c277-4f83-8acc-521d85cfa32b",
    "user" : "xxxx",
    "job_name" : "SparkSubmitTset2",
    "job_result" : "UNDEFINED",
    "job_state" : "ACCEPTED",
    "job_progress" : 0,
    "job_type" : "SparkSubmit",
    "started_time" : 0,
    "submitted_time" : 1564714020099,
    "finished_time" : 0,
    "elapsed_time" : 0,
    "queue" : "default",
    "arguments" : "[--conf, yujjsjhe, --driver-memory, yueujdjjd, --master,\nyarn-cluster,\ns3a://obs-test/
hadoop-mapreduce-examples-3.1.1.jar]",
    "launcher_id" : "application_1564622673393_0611",
    "properties" : { }
  } ]
}
```


状态码： 400

查询作业列表信息失败


```
{
  "error_msg" : "查询作业列表失败",
  "error_code" : "0166"
}
```


SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowJobExeListNewSolution {

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

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

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

```
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowJobExeListNewRequest request = new ShowJobExeListNewRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowJobExeListNewResponse response = client.showJobExeListNew(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowJobExeListNewRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_job_exe_list_new(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

```
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
202	查询作业列表信息成功
400	查询作业列表信息失败

错误码

请参见**错误码**。

6.2.3 查询单个作业信息

功能介绍

在MRS集群中查询指定作业的详细信息。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}

表 6-71 路径参数

参数	是否必选	参数类型	描述
job_execution_id	是	String	**参数解释：** 作业ID。获取方法，请参见**获取作业ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 202

表 6-72 响应 Body 参数

参数	参数类型	描述
job_detail	**JobQueryBean** object	**参数解释：** 作业详细信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-73 JobQueryBean

参数	参数类型	描述
job_id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
user	String	**参数解释：** 提交作业的用户名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_name	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_result	String	**参数解释：** 作业最终结果。 **约束限制：** 不涉及 **取值范围：** ● FAILED：执行失败的作业 ● KILLED：执行中被手动终止的作业。 ● UNDEFINED：正在执行的作业。 ● SUCCEEDED：执行成功的作业。 **默认取值：** 不涉及
job_state	String	**参数解释：** 作业执行状态。 **约束限制：** 不涉及 **取值范围：** ● FAILED：失败 ● KILLED：已终止 ● NEW：已创建 ● NEW_SAVING：已创建保存中 ● SUBMITTED：已提交 ● ACCEPTED：已接受 ● RUNNING：运行中 ● FINISHED：已完成 **默认取值：** 不涉及

参数	参数类型	描述
job_progress	Float	**参数解释：** 作业执行进度。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_type	String	**参数解释：** 作业类型。 **约束限制：** 不涉及 **取值范围：** ● MapReduce ● SparkSubmit：SparkPython类型的作业在查询时作业类型请选择SparkSubmit。 ● HiveScript ● HiveSql ● DistCp，导入、导出数据。 ● SparkScript ● SparkSql ● Flink **默认取值：** 不涉及
started_time	Long	**参数解释：** 作业开始执行时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
submitted_time	Long	**参数解释：** 作业提交时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
finished_time	Long	**参数解释：** 作业完成时间。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
elapsed_time	Long	**参数解释：** 作业执行时长。单位：毫秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
arguments	String	**参数解释：** 运行参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
launcher_id	String	**参数解释：** 真实作业编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
properties	String	**参数解释：** 配置参数，用于传-d参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
app_id	String	**参数解释：** 实际作业编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tracking_url	String	**参数解释：** 日志链接地址。当前仅SparkSubmit作业支持该参数。该参数基于集群的EIP访问集群中的YARN WebUI页面，用户如果在VPC界面解绑EIP，MRS服务侧数据会因为未更新导致该参数引用旧EIP导致访问失败，可通过对集群重新进行EIP的绑定来修复该问题。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
queue	String	**参数解释：** 作业的资源队列类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-74 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询单个作业请求示例

```
GET https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}
```

响应示例

状态码： 202

查询单个作业信息成功

```
{
  "job_detail" : {
    "job_id" : "431b135e-c090-489f-b1db-0abe3822b855",
    "user" : "xxxx",
    "job_name" : "pyspark1",
    "job_result" : "SUCCEEDED",
    "job_state" : "FINISHED",
    "job_progress" : 100,
    "job_type" : "SparkSubmit",
    "started_time" : 1564626578817,
    "submitted_time" : 1564626561541,
    "finished_time" : 1564626664930,
    "elapsed_time" : 86113,
    "queue" : "default",
    "arguments" : "[--class, org.apache.spark.examples.SparkPi, --driver-memory, 512MB, --num-executors, 1,
--executor-cores, 1, --master, yarn-cluster, s3a://obs-test/jobs/spark/spark-examples_2.11-2.1.0.jar, 10000]",
    "launcher_id" : "application_1564622673393_0006",
    "app_id" : "application_1564622673393_0007",
    "properties" : "{}"
  }
}
```

状态码： 400

查询单个作业信息失败

```
{
  "error_msg" : "查询作业失败",
  "error_code" : "0162"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowSingleJobExeSolution {

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

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

```
        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSingleJobExeRequest request = new ShowSingleJobExeRequest();
        request.withJobExecutionId("{job_execution_id}");
        request.withClusterId("{cluster_id}");
        try {
            ShowSingleJobExeResponse response = client.showSingleJobExe(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowSingleJobExeRequest()
        request.job_execution_id = "{job_execution_id}"
        request.cluster_id = "{cluster_id}"
        response = client.show_single_job_exe(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v2/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSingleJobExeRequest{}
    request.JobExecutionId = "{job_execution_id}"
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowSingleJobExe(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	查询单个作业信息成功
400	查询单个作业信息失败

错误码

请参见**错误码**。

6.2.4 终止作业

功能介绍

在MRS集群中终止指定作业。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/kill

表 6-75 路径参数

参数	是否必选	参数类型	描述
job_execution_id	是	String	**参数解释：** 作业ID。获取方法，请参见**获取作业ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

无

请求示例

终止作业请求示例

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/kill
```

响应示例

状态码： 400

在MRS集群中终止指定作业失败

```
{
  "error_msg" : "终止作业失败",
  "error_code" : "0175"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class StopJobSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        StopJobRequest request = new StopJobRequest();
        request.withJobExecutionId("{job_execution_id}");
        request.withClusterId("{cluster_id}");
        try {
            StopJobResponse response = client.stopJob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

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

    try:
        request = StopJobRequest()
        request.job_execution_id = "{job_execution_id}"
        request.cluster_id = "{cluster_id}"
        response = client.stop_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopJobRequest{}
    request.JobExecutionId = "{job_execution_id}"
    request.ClusterId = "{cluster_id}"
    response, err := client.StopJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	在MRS集群中终止指定作业成功
400	在MRS集群中终止指定作业失败

错误码

请参见**错误码**。

6.2.5 获取 SQL 结果

功能介绍

在MRS集群中查询SparkSql和SparkScript两种类型作业的SQL语句运行完成后返回的查询结果。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/sql-result

表 6-76 路径参数

参数	是否必选	参数类型	描述
job_execution_id	是	String	**参数解释：** 作业ID。获取方法，请参见**获取作业ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 202

表 6-77 响应 Body 参数

参数	参数类型	描述
sql_results	Object	**参数解释：** SQL语句查询结果。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-78 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

获取SQL结果请求示例

```
GET https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/sql-result
```

响应示例

状态码： 202

获取SQL结果成功

```
{
  "sql_results" : {
    "0" : [ {
      "result" : "succeed"
    } ],
    "1" : [ {
      "database" : "default",
      "isTemporary" : "false",
      "tableName" : "src_wordcount"
    } ],
    "2" : [ {
      "result" : "succeed"
    } ],
    "3" : [ {
      "result" : "succeed"
    } ],
    "4" : [ {
      "name" : "a",
      "id" : 1
    }, {
      "name" : "b",
      "id" : 2
    } ]
  }
}
```

状态码： 400

收集SQL作业结果失败

```
{
  "error_msg" : "收集SQL作业结果失败",
  "error_code" : "0172"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowSqlResultWithJobSolution {

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

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSqlResultWithJobRequest request = new ShowSqlResultWithJobRequest();
        request.withJobExecutionId("{job_execution_id}");
        request.withClusterId("{cluster_id}");
        try {
            ShowSqlResultWithJobResponse response = client.showSqlResultWithJob(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowSqlResultWithJobRequest()
        request.job_execution_id = "{job_execution_id}"
        request.cluster_id = "{cluster_id}"
        response = client.show_sql_result_with_job(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSqlResultWithJobRequest{}
    request.JobExecutionId = "{job_execution_id}"
    request.ClusterId = "{cluster_id}"
    response, err := client.ShowSqlResultWithJob(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	获取SQL结果成功
400	收集SQL作业结果失败

错误码

请参见**错误码**。

6.2.6 批量删除作业

功能介绍

在MRS集群中批量删除作业。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/job-executions/batch-delete

表 6-79 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-80 请求 Body 参数

参数	是否必选	参数类型	描述
job_id_list	否	Array of strings	**参数解释：** 作业ID列表。获取方法，请参见**获取作业ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

无

请求示例

批量删除作业请求示例

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/job-executions/batch-delete

{
  "job_id_list" : [ "48c45725-b699-4aa9-9bfd-f7ff87eb6fe8", "af846665-dd32-4349-a8b5-561e109c383c" ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

批量删除作业请求示例

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

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

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

public class BatchDeleteJobsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchDeleteJobsRequest request = new BatchDeleteJobsRequest();
        request.withClusterId("{cluster_id}");
        JobBatchDelete body = new JobBatchDelete();
        List<String> listbodyJobIdList = new ArrayList<>();
        listbodyJobIdList.add("48c45725-b699-4aa9-9bfd-f7ff87eb6fe8");
        listbodyJobIdList.add("af846665-dd32-4349-a8b5-561e109c383c");
        body.withJobIdList(listbodyJobIdList);
        request.withBody(body);
        try {
            BatchDeleteJobsResponse response = client.batchDeleteJobs(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
```

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

Python

批量删除作业请求示例

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchDeleteJobsRequest()
        request.cluster_id = "{cluster_id}"
        listJobIdListbody = [
            "48c45725-b699-4aa9-9bfd-f7ff87eb6fe8",
            "af846665-dd32-4349-a8b5-561e109c383c"
        ]
        request.body = JobBatchDelete(
            job_id_list=listJobIdListbody
        )
        response = client.batch_delete_jobs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

批量删除作业请求示例

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteJobsRequest{}
    request.ClusterId = "{cluster_id}"
    var listJobIdListbody = []string{
        "48c45725-b699-4aa9-9bfd-f7ff87eb6fe8",
        "af846665-dd32-4349-a8b5-561e109c383c",
    }
    request.Body = &model.JobBatchDelete{
        JobIdList: &listJobIdListbody,
    }
    response, err := client.BatchDeleteJobs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	在MRS集群中批量删除作业成功

错误码

请参见**错误码**。

6.3 弹性伸缩接口

6.3.1 查看弹性伸缩策略

功能介绍

查看指定集群的所有的弹性伸缩策略信息。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/autoscaling-policy/{cluster_id}

表 6-81 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-82 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of **AutoScalingPolicyV2** objects	**参数解释：** 弹性伸缩策略列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-83 AutoScalingPolicyV2

参数	参数类型	描述
node_group_name	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_pool_name	String	**参数解释：** 资源池名称。 **约束限制：** 当集群版本不支持按指定资源池进行弹性伸缩时，需要填写为default资源池。 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成。 **默认取值：** 不涉及

参数	参数类型	描述
auto_scaling_policy	**AutoScalingPolicyInfo** object	**参数解释：** 弹性伸缩规则。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-84 AutoScalingPolicyInfo

参数	参数类型	描述
auto_scaling_enable	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** ● true：开启自动伸缩规则 ● false：不开启自动伸缩规则 **默认取值：** 不涉及
min_capacity	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	参数类型	描述
resources_plans	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及
rules	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	Array of **Tag** objects	**参数解释：** 弹性伸缩标签列表。 **约束限制：** 不能超过20条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-85 ResourcesPlan

参数	参数类型	描述
period_type	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
start_time	String	**参数解释：** 资源计划的起始时间，格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	参数类型	描述
max_capacity	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
effective_days	Array of strings	**参数解释：** 资源计划的生效日期，为空时代表每日，另外也可为以下返回值： MONDAY（周一）、TUESDAY（周二）、WEDNESDAY（周三）、THURSDAY（周四）、FRIDAY（周五）、SATURDAY（周六）、SUNDAY（周日） **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-86 Rule

参数	参数类型	描述
name	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 在一个节点组范围内，不允许重名。 **默认取值：** 不涉及

参数	参数类型	描述
description	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度为[0-1024]个字符。 **默认取值：** 不涉及
adjustment_type	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** ● scale_out：扩容 ● scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及

参数	参数类型	描述
trigger	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-87 Trigger

参数	参数类型	描述
metric_name	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。详细指标名称内容请参见**"弹性伸缩指标列表"**。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见**"弹性伸缩指标列表"**。 **默认取值：** 不涉及
metric_value	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及

参数	参数类型	描述
comparison_operator	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** ● LT：小于 ● GT：大于 ● LTOE：小于等于 ● GTOE：大于等于 **默认取值：** 不涉及
evaluation_periods	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-288 **默认取值：** 不涉及

表 6-88 Tag

参数	参数类型	描述
key	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 同一资源的key值不能重复。 ● 最大长度128个unicode字符，不能为空字符串。 **默认取值：** 不涉及

参数	参数类型	描述
value	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 最大长度255个unicode字符，可以为空字符串。 **默认取值：** 不涉及

状态码： 400

表 6-89 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

查看弹性伸缩策略成功

```
[ {
  "node_group_name" : "task_node_analysis_group",
  "resource_pool_name" : "default",
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 0,
    "max_capacity" : 1,
    "resources_plans" : [ {
      "period_type" : "daily",
      "effective_days" : [ "SUNDAY" ],
      "start_time" : "12:00",
      "end_time" : "13:00",
      "min_capacity" : 2,
      "max_capacity" : 3
    } ],
    "rules" : [ {
      "name" : "default-expand-1",
      "description" : "",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : "100",
        "comparison_operator" : "GTOE",
        "evaluation_periods" : 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.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ShowAutoScalingPolicySolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAutoScalingPolicyRequest request = new ShowAutoScalingPolicyRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowAutoScalingPolicyResponse response = client.showAutoScalingPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAutoScalingPolicyRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_auto_scaling_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	查看弹性伸缩策略成功
400	查看弹性伸缩策略失败

错误码

请参见**错误码**。

6.3.2 更新弹性伸缩策略

功能介绍

更新弹性伸缩策略。

调用方法

请参见**如何调用API**。

URI

PUT /v2/{project_id}/autoscaling-policy/{cluster_id}

表 6-90 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-91 请求 Body 参数

参数	是否必选	参数类型	描述
node_group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_pool_name	是	String	**参数解释：** 资源池名称。 **约束限制：** 当集群版本不支持按指定资源池进行弹性伸缩时，需要填写为default资源池。 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成。 **默认取值：** 不涉及
auto_scaling_policy	否	**AutoScalingPolicyInfo** object	**参数解释：** 弹性伸缩规则。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-92 AutoScalingPolicyInfo

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** ● true：开启自动伸缩规则 ● false：不开启自动伸缩规则 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	否	Array of **Tag** objects	**参数解释：** 弹性伸缩标签列表。 **约束限制：** 不能超过20条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-93 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	**参数解释：** 资源计划的起始时间，格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
effective_days	否	Array of strings	**参数解释：** 资源计划的生效日期，为空时代表每日，另外也可为以下返回值： MONDAY（周一）、TUESDAY（周二）、WEDNESDAY（周三）、THURSDAY（周四）、FRIDAY（周五）、SATURDAY（周六）、SUNDAY（周日） **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-94 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 在一个节点组范围内，不允许重名。 **默认取值：** 不涉及
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度为[0-1024]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** • scale_out：扩容 • scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-95 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。详细指标名称内容请参见"**弹性伸缩指标列表**"。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见"**弹性伸缩指标列表**"。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** • LT：小于 • GT：大于 • LTOE：小于等于 • GTOE：大于等于 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-288 **默认取值：** 不涉及

表 6-96 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 同一资源的key值不能重复。 ● 最大长度128个unicode字符，不能为空字符串。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 最大长度255个unicode字符，可以为空字符串。 **默认取值：** 不涉及

响应参数

无

请求示例

更新一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
/v2/{project_id}/autoscaling-policy/{cluster_id}

{
  "node_group_name" : "ttt",
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 0,
    "max_capacity" : 5,
    "rules" : [ {
      "name" : "default-expand-1",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 20,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : "75",
        "comparison_operator" : "GT",
        "evaluation_periods" : 1
      }
    }, {
      "name" : "default-shrink-1",
      "adjustment_type" : "scale_in",
      "cool_down_minutes" : 20,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : "25",
        "comparison_operator" : "LT",
```

```
      "evaluation_periods" : 1
    }
  } ],
  "resources_plans" : [ {
    "period_type" : "daily",
    "start_time" : "06:00",
    "end_time" : "20:00",
    "min_capacity" : "0",
    "max_capacity" : "2",
    "effective_days" : null
  } ],
  "tags" : [ {
    "key" : "aaa",
    "value" : "bbb"
  } ]
 },
 "resource_pool_name" : "default"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

更新一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

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

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

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

public class UpdateAutoScalingPolicySolution {

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

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

```
MrsClient client = MrsClient.newBuilder()
        .withCredential(auth)
        .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
        .build();
UpdateAutoScalingPolicyRequest request = new UpdateAutoScalingPolicyRequest();
request.withClusterId("{cluster_id}");
AutoScalingPolicyV2 body = new AutoScalingPolicyV2();
List<Tag> listAutoScalingPolicyTags = new ArrayList<>();
listAutoScalingPolicyTags.add(
    new Tag()
        .withKey("aaa")
        .withValue("bbb")
);
Trigger triggerRules = new Trigger();
triggerRules.withMetricName("YARNAppRunning")
    .withMetricValue("25")
    .withComparisonOperator("LT")
    .withEvaluationPeriods(1);
Trigger triggerRules1 = new Trigger();
triggerRules1.withMetricName("YARNAppRunning")
    .withMetricValue("75")
    .withComparisonOperator("GT")
    .withEvaluationPeriods(1);
List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-expand-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
        .withCoolDownMinutes(20)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules1)
);
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-shrink-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_in"))
        .withCoolDownMinutes(20)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules)
);
List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("06:00")
        .withEndTime("20:00")
        .withMinCapacity(0)
        .withMaxCapacity(2)
);
AutoScalingPolicyInfo autoScalingPolicybody = new AutoScalingPolicyInfo();
autoScalingPolicybody.withAutoScalingEnable(true)
    .withMinCapacity(0)
    .withMaxCapacity(5)
    .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
    .withRules(listAutoScalingPolicyRules)
    .withTags(listAutoScalingPolicyTags);
body.withAutoScalingPolicy(autoScalingPolicybody);
body.withResourcePoolName("default");
body.withNodeGroupName("ttt");
request.withBody(body);
try {
    UpdateAutoScalingPolicyResponse response = client.updateAutoScalingPolicy(request);
    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

更新一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAutoScalingPolicyRequest()
        request.cluster_id = "{cluster_id}"
        listTagsAutoScalingPolicy = [
            Tag(
                key="aaa",
                value="bbb"
            )
        ]
        triggerRules = Trigger(
            metric_name="YARNAppRunning",
            metric_value="25",
            comparison_operator="LT",
            evaluation_periods=1
        )
        triggerRules1 = Trigger(
            metric_name="YARNAppRunning",
            metric_value="75",
            comparison_operator="GT",
            evaluation_periods=1
        )
        listRulesAutoScalingPolicy = [
            Rule(
                name="default-expand-1",
                adjustment_type="scale_out",
                cool_down_minutes=20,
```

```
            scaling_adjustment=1,
            trigger=triggerRules1
        ),
        Rule(
            name="default-shrink-1",
            adjustment_type="scale_in",
            cool_down_minutes=20,
            scaling_adjustment=1,
            trigger=triggerRules
        )
    ]
    listResourcesPlansAutoScalingPolicy = [
        ResourcesPlan(
            period_type="daily",
            start_time="06:00",
            end_time="20:00",
            min_capacity=0,
            max_capacity=2
        )
    ]
    autoScalingPolicybody = AutoScalingPolicyInfo(
        auto_scaling_enable=True,
        min_capacity=0,
        max_capacity=5,
        resources_plans=listResourcesPlansAutoScalingPolicy,
        rules=listRulesAutoScalingPolicy,
        tags=listTagsAutoScalingPolicy
    )
    request.body = AutoScalingPolicyV2(
        auto_scaling_policy=autoScalingPolicybody,
        resource_pool_name="default",
        node_group_name="ttt"
    )
    response = client.update_auto_scaling_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

更新一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ， 并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ， 并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
package main

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

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

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

client := mrs.NewMrsClient(
    mrs.MrsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateAutoScalingPolicyRequest{}
request.ClusterId = "{cluster_id}"
var listTagsAutoScalingPolicy = []model.Tag{
    {
        Key: "aaa",
        Value: "bbb",
    },
}
comparisonOperatorTrigger:= "LT"
triggerRules := &model.Trigger{
    MetricName: "YARNAppRunning",
    MetricValue: "25",
    ComparisonOperator: &comparisonOperatorTrigger,
    EvaluationPeriods: int32(1),
}
comparisonOperatorTrigger1:= "GT"
triggerRules1 := &model.Trigger{
    MetricName: "YARNAppRunning",
    MetricValue: "75",
    ComparisonOperator: &comparisonOperatorTrigger1,
    EvaluationPeriods: int32(1),
}
var listRulesAutoScalingPolicy = []model.Rule{
    {
        Name: "default-expand-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
        CoolDownMinutes: int32(20),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules1,
    },
    {
        Name: "default-shrink-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_IN,
        CoolDownMinutes: int32(20),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules,
    },
}
var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
    {
        PeriodType: "daily",
        StartTime: "06:00",
        EndTime: "20:00",
        MinCapacity: int32(0),
        MaxCapacity: int32(2),
    },
}
autoScalingPolicybody := &model.AutoScalingPolicyInfo{
    AutoScalingEnable: true,
    MinCapacity: int32(0),
    MaxCapacity: int32(5),
    ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
    Rules: &listRulesAutoScalingPolicy,
    Tags: &listTagsAutoScalingPolicy,
}
request.Body = &model.AutoScalingPolicyV2{
```

```
        AutoScalingPolicy: autoScalingPolicybody,
        ResourcePoolName: "default",
        NodeGroupName: "ttt",
    }
    response, err := client.UpdateAutoScalingPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	更新弹性伸缩策略成功

错误码

请参见**错误码**。

6.3.3 删除弹性伸缩策略

功能介绍

删除弹性伸缩策略。

调用方法

请参见**如何调用API**。

URI

DELETE /v2/{project_id}/autoscaling-policy/{cluster_id}

表 6-97 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-98 请求 Body 参数

参数	是否必选	参数类型	描述
node_group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 如果resource_pool_name为default，则删除节点组维度的弹性伸缩策略。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
resource_pool_name	是	String	**参数解释：** 资源池名称。 **约束限制：** 当集群版本不支持按指定资源池进行弹性伸缩时，需要填写为default资源池。 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成。 **默认取值：** 不涉及

响应参数

无

请求示例

- 删除node_group_2节点组维度策略

```
/v2/174ee662a7e24cc99bfc858c4558dbf6/autoscaling-policy/daf42ff0-05bc-4a1e-afbf-42b3131a1295

{
  "node_group_name" : "node_group_2",
  "resource_pool_name" : "default"
}
```

- 删除节点组策略里的resource_1资源池策略

```
/v2/174ee662a7e24cc99bfc858c4558dbf6/autoscaling-policy/daf42ff0-05bc-4a1e-afbf-42b3131a1295

{
  "node_group_name" : "node_group_2",
  "resource_pool_name" : "resource_1"
}
```

响应示例

无

状态码

状态码	描述
202	删除弹性伸缩策略成功

错误码

请参见**错误码**。

6.3.4 创建弹性伸缩策略

功能介绍

创建弹性伸缩策略。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/autoscaling-policy/{cluster_id}

表 6-99 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-100 请求 Body 参数

参数	是否必选	参数类型	描述
node_group_name	是	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_pool_name	是	String	**参数解释：** 资源池名称。 **约束限制：** 当集群版本不支持按指定资源池进行弹性伸缩时，需要填写为default资源池。 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成。 **默认取值：** 不涉及
auto_scaling_policy	否	**AutoScalingPolicyInfo** object	**参数解释：** 弹性伸缩规则。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-101 AutoScalingPolicyInfo

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** ● true：开启自动伸缩规则 ● false：不开启自动伸缩规则 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	否	Array of **Tag** objects	**参数解释：** 弹性伸缩标签列表。 **约束限制：** 不能超过20条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-102 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
start_time	是	String	**参数解释：** 资源计划的起始时间，格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
effective_days	否	Array of strings	**参数解释：** 资源计划的生效日期，为空时代表每日，另外也可为以下返回值： MONDAY（周一）、TUESDAY（周二）、WEDNESDAY（周三）、THURSDAY（周四）、FRIDAY（周五）、SATURDAY（周六）、SUNDAY（周日） **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-103 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 在一个节点组范围内，不允许重名。 **默认取值：** 不涉及
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度为[0-1024]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** • scale_out：扩容 • scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-104 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。详细指标名称内容请参见**"弹性伸缩指标列表"**。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见**"弹性伸缩指标列表"**。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** • LT：小于 • GT：大于 • LTOE：小于等于 • GTOE：大于等于 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-288 **默认取值：** 不涉及

表 6-105 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 同一资源的key值不能重复。 ● 最大长度128个unicode字符，不能为空字符串。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 ● 最大长度255个unicode字符，可以为空字符串。 **默认取值：** 不涉及

响应参数

无

请求示例

创建一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
/v2/{project_id}/autoscaling-policy/{cluster_id}

{
  "node_group_name" : "node_group_1",
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 0,
    "max_capacity" : 5,
    "rules" : [ {
      "name" : "default-expand-1",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 20,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : "75",
        "comparison_operator" : "GT",
        "evaluation_periods" : 1
      }
    }, {
      "name" : "default-shrink-1",
      "adjustment_type" : "scale_in",
      "cool_down_minutes" : 20,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNAppRunning",
        "metric_value" : "25",
        "comparison_operator" : "LT",
```

```
      "evaluation_periods" : 1
    }
  } ],
  "resources_plans" : [ {
    "period_type" : "daily",
    "start_time" : "06:00",
    "end_time" : "20:00",
    "min_capacity" : "0",
    "max_capacity" : "2",
    "effective_days" : [ "MONDAY" ]
  } ],
  "tags" : [ {
    "key" : "aaa",
    "value" : "bbb"
  } ]
 },
 "resource_pool_name" : "default"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

创建一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

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

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

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

public class CreateAutoScalingPolicySolution {

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

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

```
MrsClient client = MrsClient.newBuilder()
        .withCredential(auth)
        .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
        .build();
CreateAutoScalingPolicyRequest request = new CreateAutoScalingPolicyRequest();
request.withClusterId("{cluster_id}");
AutoScalingPolicyV2 body = new AutoScalingPolicyV2();
List<Tag> listAutoScalingPolicyTags = new ArrayList<>();
listAutoScalingPolicyTags.add(
    new Tag()
        .withKey("aaa")
        .withValue("bbb")
);
Trigger triggerRules = new Trigger();
triggerRules.withMetricName("YARNAppRunning")
    .withMetricValue("25")
    .withComparisonOperator("LT")
    .withEvaluationPeriods(1);
Trigger triggerRules1 = new Trigger();
triggerRules1.withMetricName("YARNAppRunning")
    .withMetricValue("75")
    .withComparisonOperator("GT")
    .withEvaluationPeriods(1);
List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-expand-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
        .withCoolDownMinutes(20)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules1)
);
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-shrink-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_in"))
        .withCoolDownMinutes(20)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules)
);
List<ResourcesPlan.EffectiveDaysEnum> listResourcesPlansEffectiveDays = new ArrayList<>();
listResourcesPlansEffectiveDays.add(ResourcesPlan.EffectiveDaysEnum.fromValue("MONDAY"));
List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("06:00")
        .withEndTime("20:00")
        .withMinCapacity(0)
        .withMaxCapacity(2)
        .withEffectiveDays(listResourcesPlansEffectiveDays)
);
AutoScalingPolicyInfo autoScalingPolicybody = new AutoScalingPolicyInfo();
autoScalingPolicybody.withAutoScalingEnable(true)
    .withMinCapacity(0)
    .withMaxCapacity(5)
    .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
    .withRules(listAutoScalingPolicyRules)
    .withTags(listAutoScalingPolicyTags);
body.withAutoScalingPolicy(autoScalingPolicybody);
body.withResourcePoolName("default");
body.withNodeGroupName("node_group_1");
request.withBody(body);
try {
    CreateAutoScalingPolicyResponse response = client.createAutoScalingPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
```

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

Python

创建一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ，并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ，并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateAutoScalingPolicyRequest()
        request.cluster_id = "{cluster_id}"
        listTagsAutoScalingPolicy = [
            Tag(
                key="aaa",
                value="bbb"
            )
        ]
        triggerRules = Trigger(
            metric_name="YARNAppRunning",
            metric_value="25",
            comparison_operator="LT",
            evaluation_periods=1
        )
        triggerRules1 = Trigger(
            metric_name="YARNAppRunning",
            metric_value="75",
            comparison_operator="GT",
            evaluation_periods=1
        )
        listRulesAutoScalingPolicy = [
            Rule(
```

```
            name="default-expand-1",
            adjustment_type="scale_out",
            cool_down_minutes=20,
            scaling_adjustment=1,
            trigger=triggerRules1
        ),
        Rule(
            name="default-shrink-1",
            adjustment_type="scale_in",
            cool_down_minutes=20,
            scaling_adjustment=1,
            trigger=triggerRules
        )
    ]
    listEffectiveDaysResourcesPlans = [
        "MONDAY"
    ]
    listResourcesPlansAutoScalingPolicy = [
        ResourcesPlan(
            period_type="daily",
            start_time="06:00",
            end_time="20:00",
            min_capacity=0,
            max_capacity=2,
            effective_days=listEffectiveDaysResourcesPlans
        )
    ]
    autoScalingPolicybody = AutoScalingPolicyInfo(
        auto_scaling_enable=True,
        min_capacity=0,
        max_capacity=5,
        resources_plans=listResourcesPlansAutoScalingPolicy,
        rules=listRulesAutoScalingPolicy,
        tags=listTagsAutoScalingPolicy
    )
    request.body = AutoScalingPolicyV2(
        auto_scaling_policy=autoScalingPolicybody,
        resource_pool_name="default",
        node_group_name="node_group_1"
    )
    response = client.create_auto_scaling_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

创建一个弹性伸缩策略， 在节点组node_group_1中配置节点组数量范围为0-5，并配置2个伸缩规则。 如果YARNAppRunning 大于等于 75 ， 并持续 1 个五分钟，则添加 1 个Task节点。如果 YARNAppRunning 小于等于25 ， 并持续 1 个五分钟，则终止 1 个Task节点。冷却时间皆为20分钟，并且设置了该策略下弹出的节点会打上aaa=bbb的标签。

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAutoScalingPolicyRequest{}
    request.ClusterId = "{cluster_id}"
    var listTagsAutoScalingPolicy = []model.Tag{
        {
            Key: "aaa",
            Value: "bbb",
        },
    }
    comparisonOperatorTrigger:= "LT"
    triggerRules := &model.Trigger{
        MetricName: "YARNAppRunning",
        MetricValue: "25",
        ComparisonOperator: &comparisonOperatorTrigger,
        EvaluationPeriods: int32(1),
    }
    comparisonOperatorTrigger1:= "GT"
    triggerRules1 := &model.Trigger{
        MetricName: "YARNAppRunning",
        MetricValue: "75",
        ComparisonOperator: &comparisonOperatorTrigger1,
        EvaluationPeriods: int32(1),
    }
    var listRulesAutoScalingPolicy = []model.Rule{
        {
            Name: "default-expand-1",
            AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
            CoolDownMinutes: int32(20),
            ScalingAdjustment: int32(1),
            Trigger: triggerRules1,
        },
        {
            Name: "default-shrink-1",
            AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_IN,
            CoolDownMinutes: int32(20),
            ScalingAdjustment: int32(1),
            Trigger: triggerRules,
        },
    }
    var listEffectiveDaysResourcesPlans = []model.ResourcesPlanEffectiveDays{
        model.GetResourcesPlanEffectiveDaysEnum().MONDAY,
    }
    var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
        {
            PeriodType: "daily",
            StartTime: "06:00",
            EndTime: "20:00",
            MinCapacity: int32(0),
            MaxCapacity: int32(2),
            EffectiveDays: &listEffectiveDaysResourcesPlans,
```

```
        },
    }
    autoScalingPolicybody := &model.AutoScalingPolicyInfo{
        AutoScalingEnable: true,
        MinCapacity: int32(0),
        MaxCapacity: int32(5),
        ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
        Rules: &listRulesAutoScalingPolicy,
        Tags: &listTagsAutoScalingPolicy,
    }
    request.Body = &model.AutoScalingPolicyV2{
        AutoScalingPolicy: autoScalingPolicybody,
        ResourcePoolName: "default",
        NodeGroupName: "node_group_1",
    }
    response, err := client.CreateAutoScalingPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	创建弹性伸缩策略完成

错误码

请参见**错误码**。

6.4 集群 HDFS 文件接口

6.4.1 获取指定目录文件列表

功能介绍

在MRS集群中获取指定目录文件列表。

接口约束

无

调用方法

请参见**如何调用API**。

URI

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

表 6-106 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 6-107 Query 参数

参数	是否必选	参数类型	描述
path	是	String	**参数解释：** 文件目录。比如访问“/tmp/test”目录列表，此处必须是目录，整体URI为/v2/{project_id}/clusters/{cluster_id}/files?path=%2Ftmp%2Ftest。 **约束限制：** 不涉及 **取值范围：** 单层目录要遵循以下规则： ● 不能为空 ● 不能以"."开头或结尾 ● 不能包括下列符号 : :*?"<>|;&,'`!{}[]$%+ ● 不能超过255个字节 **默认取值：** 不涉及
offset	否	String	**参数解释：** 分页参数，表示从该偏移量开始查询文件列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 1
limit	否	String	**参数解释：** 分页参数，列表当前分页的数量限制。 **约束限制：** 不涉及 **取值范围：** ≤1000 **默认取值：** 100

参数	是否必选	参数类型	描述
sort_key	否	String	**参数解释：** 列表排序按该属性排序。 **约束限制：** 不涉及 **取值范围：** ● path_suffix：文件或目录名称 ● length：文件大小 ● modification_time：修改时间 **默认取值：** path_suffix
order	否	String	**参数解释：** 列表排序方式。 **约束限制：** 不涉及 **取值范围：** ● desc：降序排列 ● asc：升序排列 **默认取值：** desc

请求参数

无

响应参数

状态码： 200

表 6-108 响应 Body 参数

参数	参数类型	描述
total_count	Long	**参数解释：** 文件总数，与分页无关。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
files	Array of **FileStatusV2** objects	**参数解释：** 文件列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-109 FileStatusV2

参数	参数类型	描述
path_suffix	String	**参数解释：** 文件在当前目录下的后缀，如获取“/tmp”目录，下面的“/tmp/test”文件，此处path_suffix内容为“test”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
owner	String	**参数解释：** 文件拥有者。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
group	String	**参数解释：** 文件属组。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
permission	String	**参数解释：** 权限信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
replication	Integer	**参数解释：** 副本数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
block_size	Integer	**参数解释：** 块大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
length	Integer	**参数解释：** 文件长度。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
type	String	**参数解释：** 文件类型。 **约束限制：** 不涉及 **取值范围：** ● FILE：文件 ● DIRECTORY：目录 **默认取值：** 不涉及
children_num	Integer	**参数解释：** 该目录下的文件条目数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
access_time	Long	**参数解释：** 文件访问时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
modification_time	Long	**参数解释：** 文件修改时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

获取指定目录文件列表请求示例

```
GET /v2/{project_id}/clusters/{cluster_id}/files?
path={directory}&offset={offset}&limit={limit}&sort_key={sort_key}&order={order}
```

响应示例

状态码： 200

获取指定目录文件列表成功

```
{
  "total_count" : 2,
  "files" : [ {
    "access_time" : 0,
    "block_size" : 0,
    "children_num" : 0,
    "group" : "hadoop",
    "length" : 0,
    "modification_time" : 1587179516623,
    "owner" : "hdfs",
    "path_suffix" : "app-logs",
    "permission" : 777,
    "replication" : 0,
    "type" : "DIRECTORY"
  }, {
    "access_time" : 1587267212761,
    "block_size" : 134217728,
    "children_num" : 0,
    "group" : "hadoop",
    "length" : 23666188,
    "modification_time" : 1587222156003,
```

```
    "owner" : "root",
    "path_suffix" : "data-m-00000",
    "permission" : "644",
    "replication" : 3,
    "type" : "FILE"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowHdfsFileListSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowHdfsFileListRequest request = new ShowHdfsFileListRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowHdfsFileListResponse response = client.showHdfsFileList(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowHdfsFileListRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_hdfs_file_list(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	获取指定目录文件列表成功

错误码

请参见**错误码**。

6.5 SQL 接口

6.5.1 提交 SQL 语句

功能介绍

在MRS集群中提交并执行一条SQL语句。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/sql-execution

表 6-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-111 请求 Body 参数

参数	是否必选	参数类型	描述
sql_type	是	String	**参数解释：** SQL类型。目前仅支持“presto”类型的SQL。 **约束限制：** ● 只有包含Presto组件的集群才能提交执行presto类型的SQL。 ● 当前仅MRS 2.0.6版本的MRS 2.0.6.1补丁、MRS 2.1.0版本的MRS 2.1.0.7补丁、MRS 3.1.2及之后版本集群支持。 **取值范围：** 不涉及 **默认取值：** 不涉及
sql_content	是	String	**参数解释：** 待执行的SQL语句。目前仅支持执行单条语句，语句中不包含“;”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
database	否	String	**参数解释：** 执行SQL所在的数据库。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** default

参数	是否必选	参数类型	描述
archive_path	否	String	**参数解释：** SQL执行结果的转储文件夹。只有select语句才会转储查询的结果。当前仅支持转储到OBS中。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-112 响应 Body 参数

参数	参数类型	描述
id	String	**参数解释：** SQL的执行id。执行select、show和desc语句时才会生成id，其他操作id为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
message	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
statement	String	**参数解释：** 执行的SQL语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	String	**参数解释：** SQL的执行状态。 **约束限制：** 不涉及 **取值范围：** ● QUEUED：正在排队 ● WAITING_FOR_RESOURCES：正在等待资源 ● PLANNING：正在规划 ● STARTING：正在启动 ● RUNNING：正在运行 ● FINISHING：即将完成 ● FINISHED：已完成 ● FAILED：执行失败 **默认取值：** 不涉及
result_location	String	**参数解释：** SQL查询语句的最终结果归档路径。只有select的语句才会在将SQL的执行结果转储到result_location中。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
content	Array<Array<String>>	**参数解释：** SQL的执行结果。只有非select的语句才会在content中返回结果，如果SQL中没有结果，content为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-113 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

提交一个presto类型的SQL语句

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/sql-execution

{
```

```
  "sql_type" : "presto",
  "sql_content" : "show tables",
  "database" : "default",
  "archive_path" : "obs://my-bucket/path"
}
```

响应示例

状态码： 200

提交SQL语句成功

```
{
  "id" : "20190909_011820_00151_xxxxx",
  "statement" : "show tables",
  "status" : "FINISHED",
  "result_location" : " obs://my_bucket/uuid_date/xxxx.csv",
  "content" : [ [ "t1", null ], [ null, "t2" ], [ null, "t3" ] ]
}
```

状态码： 400

提交SQL语句失败

```
{
  "error_code" : "MRS.0011",
  "message" : "提交SQL到Executor上失败，集群ID为xxxx"
}
```

状态码

状态码	描述
200	提交SQL语句成功
400	提交SQL语句失败

错误码

请参见**错误码**。

6.5.2 查询 SQL 结果

功能介绍

在MRS集群中查询一条SQL的执行结果。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}

表 6-114 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
sql_id	是	String	**参数解释：** SQL的执行ID，即提交SQL语句返回结果中的sql_id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-115 响应 Body 参数

参数	参数类型	描述
id	String	**参数解释：** SQL的执行id。执行select、show和desc语句时才会生成id，其他操作id为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
message	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
statement	String	**参数解释：** 执行的SQL语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
status	String	**参数解释：** SQL的执行状态。 **约束限制：** 不涉及 **取值范围：** ● QUEUED：正在排队 ● WAITING_FOR_RESOURCES：正在等待资源 ● PLANNING：正在规划 ● STARTING：正在启动 ● RUNNING：正在运行 ● FINISHING：即将完成 ● FINISHED：已完成 ● FAILED：执行失败 **默认取值：** 不涉及
result_location	String	**参数解释：** SQL查询语句的最终结果归档路径。只有select的语句才会在将SQL的执行结果转储到result_location中。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
content	Array<Array<String>>	**参数解释：** SQL的执行结果。只有非select的语句才会在content中返回结果，如果SQL中没有结果，content为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-116 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询SQL结果请求示例

```
GET https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}
```

响应示例

状态码： 200

查询SQL结果成功

```
{
  "id" : "20190909_011820_00151_xxxxx",
  "statement" : "show tables",
  "status" : "FINISHED",
  "result_location" : "obs://my_bucket/uuid_date/xxxx.csv",
  "content" : [ [ "t1", null ], [ null, "t2" ], [ null, "t3" ] ]
}
```

状态码： 400

查询SQL结果失败

```
{
  "error_code" : "MRS.0011",
  "message" : "提交SQL到Executor上失败，集群ID为xxxx"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowSqlResultSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSqlResultRequest request = new ShowSqlResultRequest();
        request.withClusterId("{cluster_id}");
        request.withSqlId("{sql_id}");
        try {
            ShowSqlResultResponse response = client.showSqlResult(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowSqlResultRequest()
        request.cluster_id = "{cluster_id}"
        request.sql_id = "{sql_id}"
        response = client.show_sql_result(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSqlResultRequest{}
    request.ClusterId = "{cluster_id}"
    request.SqlId = "{sql_id}"
    response, err := client.ShowSqlResult(request)
    if err == nil {
```

```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，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结果成功
400	查询SQL结果失败

错误码

请参见**错误码**。

6.5.3 取消 SQL 执行任务

功能介绍

在MRS集群中取消一条SQL的执行任务。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}/cancel

表 6-117 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
sql_id	是	String	**参数解释：** SQL的执行ID，即提交SQL语句返回结果中的sql_id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-118 响应 Body 参数

参数	参数类型	描述
message	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	String	**参数解释：** 取消SQL的执行结果。默认返回SUCCEED，对于已经结束的任务也会返回SUCCEED，只有取消正在运行的SQL时没成功才会FAILED。 **约束限制：** 不涉及 **取值范围：** ● SUCCEED：成功 ● FAILED：失败 **默认取值：** 不涉及

状态码： 400

表 6-119 响应 Body 参数

参数	参数类型	描述
message	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
status	String	**参数解释：** 取消SQL的执行结果。默认返回SUCCEED，对于已经结束的任务也会返回SUCCEED，只有取消正在运行的SQL时没成功才会FAILED。 **约束限制：** 不涉及 **取值范围：** ● SUCCEED：成功 ● FAILED：失败 **默认取值：** 不涉及

请求示例

取消SQL执行任务请求示例

```
POST https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}/cancel
```

响应示例

状态码： 200

取消SQL执行任务成功

```
{
  "status" : "SUCCEED"
}
```

状态码： 400

取消SQL执行任务失败

```
{
  "status" : "FAILED",
  "message" : "Cancel sql error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
```

```
import com.huaweicloud.sdk.mrs.v2.model.*;


public class CancelSqlSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CancelSqlRequest request = new CancelSqlRequest();
        request.withClusterId("{cluster_id}");
        request.withSqlId("{sql_id}");
        try {
            CancelSqlResponse response = client.cancelSql(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

    client = MrsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(MrsRegion.value_of("<YOUR REGION>")) \
```

```
        .build()

    try:
        request = CancelSqlRequest()
        request.cluster_id = "{cluster_id}"
        request.sql_id = "{sql_id}"
        response = client.cancel_sql(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CancelSqlRequest{}
    request.ClusterId = "{cluster_id}"
    request.SqlId = "{sql_id}"
    response, err := client.CancelSql(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	取消SQL执行任务成功
400	取消SQL执行任务失败

错误码

请参见**错误码**。

6.6 委托管理

6.6.1 查询用户（组）与 IAM 委托的映射关系

功能介绍

获取用户（组）与IAM委托之间的映射关系的详细信息。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/agency-mapping

表 6-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-121 响应 Body 参数

参数	参数类型	描述
agency_mappings	Array of **AgencyMapping** objects	**参数解释：** 用户（组）与委托之间的映射关系详细信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-122 AgencyMapping

参数	参数类型	描述
agency	String	**参数解释：** 该映射绑定的IAM委托名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
identifier_type	String	**参数解释：** 委托类型，分为“User”和“Group”两种。 **约束限制：** 不涉及 **取值范围：** ● User：表示该映射关系为针对用户的映射，identifiers中填写用户名称列表。 ● Group：表示该映射关系为针对用户组的映射，identifiers中填写用户组名称列表。 **默认取值：** 不涉及
identifiers	Array of strings	**参数解释：** IAM委托映射的用户（组）名称列表。请前往IAM，单击“用户（组）”按钮，获取用户（组）名称列表。获取方法，请参见**获取委托用户组名称**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
agency_id	String	**参数解释：** 该映射关系绑定的委托的唯一标识码。请前往IAM，单击“委托”按钮，进入委托页面，将鼠标放置委托名称上，在弹窗中获取委托唯一标识码。获取方法，请参见**获取委托名称和ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

状态码： 400

表 6-123 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

查询用户（组）与IAM委托的映射关系成功

```
{
  "agency_mappings" : [ {
    "agency" : "agency01",
    "identifier_type" : "User",
    "identifiers" : [ "user01" ],
    "agency_id" : "092adc623c00d2ea4fdac01d4b637f0b"
  }, {
    "agency" : "agency02",
    "identifier_type" : "User",
    "identifiers" : [ "user02" ],
    "agency_id" : "065239307e00d3ae4f80c01d4bdafdfd"
  }, {
    "agency" : "groupAgency",
    "identifier_type" : "Group",
    "identifiers" : [ "group01", "group02", "group03" ],
    "agency_id" : "08467a446200d5ac4ff9c01d56670c3b"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowAgencyMappingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAgencyMappingRequest request = new ShowAgencyMappingRequest();
        request.withClusterId("{cluster_id}");
        try {
```

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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAgencyMappingRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_agency_mapping(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
   // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
   ak := os.Getenv("CLOUD_SDK_AK")
   sk := os.Getenv("CLOUD_SDK_SK")
   projectId := "{project_id}"

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

   client := mrs.NewMrsClient(
      mrs.MrsClientBuilder().
         WithRegion(region.ValueOf("<YOUR REGION>")).
         WithCredential(auth).
         Build())

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

更多

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

状态码

状态码	描述
200	查询用户（组）与IAM委托的映射关系成功
400	查询用户（组）与IAM委托的映射关系失败

错误码

请参见**错误码**。

6.6.2 更新用户（组）与 IAM 委托的映射关系

功能介绍

更新用户（组）与IAM委托之间的映射关系。

接口约束

无

调用方法

请参见**如何调用API**。

URI

PUT /v2/{project_id}/clusters/{cluster_id}/agency-mapping

表 6-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-125 请求 Body 参数

参数	是否必选	参数类型	描述
agency_mappings	是	Array of **AgencyMapping** objects	**参数解释：** 用户（组）与委托之间的映射关系详细信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-126 AgencyMapping

参数	是否必选	参数类型	描述
agency	是	String	**参数解释：** 该映射绑定的IAM委托名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
identifier_type	是	String	**参数解释：** 委托类型，分为“User”和“Group”两种。 **约束限制：** 不涉及 **取值范围：** ● User：表示该映射关系为针对用户的映射，identifiers中填写用户名称列表。 ● Group：表示该映射关系为针对用户组的映射，identifiers中填写用户组名称列表。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
identifiers	是	Array of strings	**参数解释：** IAM委托映射的用户（组）名称列表。请前往IAM，单击“用户（组）”按钮，获取用户（组）名称列表。获取方法，请参见**获取委托用户组名称**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
agency_id	是	String	**参数解释：** 该映射关系绑定的委托的唯一标识码。请前往IAM，单击“委托”按钮，进入委托页面，将鼠标放置委托名称上，在弹窗中获取委托唯一标识码。获取方法，请参见**获取委托名称和ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-127 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 更新映射请求操作结果。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed：操作失败 **默认取值：** 不涉及

状态码： 400

表 6-128 响应 Body 参数

参数	参数类型	描述
error_code	String	**参数解释：** 错误码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
error_msg	String	**参数解释：** 错误描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

更新用户（组）与IAM委托的映射关系示例

```
PUT https://{endpoint}/v2/{project_id}/clusters/{cluster_id}/agency-mapping
```

```
{
  "agency_mappings" : [ {
    "agency" : "agency01",
    "identifier_type" : "User",
    "identifiers" : [ "test" ],
    "agency_id" : "xxxx"
  } ]
}
```

响应示例

状态码： 200

更新用户（组）与IAM委托的映射关系成功

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新用户（组）与IAM委托的映射关系示例

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

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

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

public class UpdateAgencyMappingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateAgencyMappingRequest request = new UpdateAgencyMappingRequest();
        request.withClusterId("{cluster_id}");
        AgencyMappingArray body = new AgencyMappingArray();
        List<String> listAgencyMappingsIdentifiers = new ArrayList<>();
```

```
        listAgencyMappingsIdentifiers.add("test");
        List<AgencyMapping> listbodyAgencyMappings = new ArrayList<>();
        listbodyAgencyMappings.add(
            new AgencyMapping()
                .withAgency("agency01")
                .withIdentifierType("User")
                .withIdentifiers(listAgencyMappingsIdentifiers)
                .withAgencyId("xxxx")
        );
        body.withAgencyMappings(listbodyAgencyMappings);
        request.withBody(body);
        try {
            UpdateAgencyMappingResponse response = client.updateAgencyMapping(request);
            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

更新用户（组）与IAM委托的映射关系示例

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAgencyMappingRequest()
        request.cluster_id = "{cluster_id}"
        listIdentifiersAgencyMappings = [
            "test"
        ]
        listAgencyMappingsbody = [
            AgencyMapping(
                agency="agency01",
                identifier_type="User",
                identifiers=listIdentifiersAgencyMappings,
                agency_id="xxxx"
```

```
            )
        ]
        request.body = AgencyMappingArray(
            agency_mappings=listAgencyMappingsbody
        )
        response = client.update_agency_mapping(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新用户（组）与IAM委托的映射关系示例

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAgencyMappingRequest{}
    request.ClusterId = "{cluster_id}"
    var listIdentifiersAgencyMappings = []string{
        "test",
    }
    var listAgencyMappingsbody = []model.AgencyMapping{
        {
            Agency: "agency01",
            IdentifierType: "User",
            Identifiers: listIdentifiersAgencyMappings,
            AgencyId: "xxxx",
        },
    }
    request.Body = &model.AgencyMappingArray{
        AgencyMappings: listAgencyMappingsbody,
    }
    response, err := client.UpdateAgencyMapping(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

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

更多

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

状态码

状态码	描述
200	更新用户（组）与IAM委托的映射关系成功
400	更新用户（组）与IAM委托的映射关系失败

错误码

请参见**错误码**。

6.7 数据连接管理

6.7.1 创建数据连接

功能介绍

创建数据连接

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/data-connectors

表 6-129 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-130 请求 Body 参数

参数	是否必选	参数类型	描述
data_connector	是	**DataConnector** object	**参数解释：** 数据连接。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-131 DataConnector

参数	是否必选	参数类型	描述
connector_name	是	String	**参数解释：** 数据连接名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
source_type	是	String	**参数解释：** 数据连接类型。 **约束限制：** 不涉及 **取值范围：** ● RDS_POSTGRES：RDS服务PostgreSQL数据库 ● RDS_MYSQL：RDS服务MySQL数据库 ● gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及
source_info	是	String	**参数解释：** 数据源信息，为json格式。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 201

表 6-132 响应 Body 参数

参数	参数类型	描述
connector_id	String	**参数解释：** 数据连接创建成功后系统返回的数据连接ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

创建mysql数据连接

```
/v2/ac66f1a5-e8f4-4399-8ec6-2c8cb1aefda7/data-connector

{
  "data_connector" : {
    "connector_name" : "mrs-mysql-connector",
    "source_type" : "RDS_MYSQL",
    "source_info" : "{\"db_name\":\"default\",\"user_name\":\"mrs\",\"password\":\"*****\",\"rds_instance_id
\":\"3aa55f1cb4c3491686936130f21e9f16in01\"}"
  }
}
```

响应示例

状态码： 201

数据连接创建成功

```
{
  "connector_id" : "0822239c1e80d4502f82c008937da9b5"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建mysql数据连接

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

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


public class CreateDataConnectorSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
```

```
        CreateDataConnectorRequest request = new CreateDataConnectorRequest();
        DataConnectorReq body = new DataConnectorReq();
        DataConnector dataConnectorbody = new DataConnector();
        dataConnectorbody.withConnectorName("mrs-mysql-connector")
            .withSourceType(DataConnector.SourceTypeEnum.fromValue("RDS_MYSQL"))
            .withSourceInfo("{"db_name":"default","user_name":"mrs","password":"*****","rds_instance_id":"3aa5
5f1cb4c3491686936130f21e9f16in01"}");
        body.withDataConnector(dataConnectorbody);
        request.withBody(body);
        try {
            CreateDataConnectorResponse response = client.createDataConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建mysql数据连接

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateDataConnectorRequest()
        dataConnectorbody = DataConnector(
            connector_name="mrs-mysql-connector",
            source_type="RDS_MYSQL",

source_info="{"db_name":"default","user_name":"mrs","password":"*****","rds_instance_id":"3aa55f1cb4c349
1686936130f21e9f16in01"}"
        )
        request.body = DataConnectorReq(
            data_connector=dataConnectorbody
        )
        response = client.create_data_connector(request)
        print(response)
```

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

Go

创建mysql数据连接

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDataConnectorRequest{}
    dataConnectorbody := &model.DataConnector{
        ConnectorName: "mrs-mysql-connector",
        SourceType: model.GetDataConnectorSourceTypeEnum().RDS_MYSQL,
        SourceInfo: "{"db_name":"default","user_name":"mrs","password":"*****","rds_instance_id":"3aa55f1cb4c3491686936130f21e9f16in01"}",
    }
    request.Body = &model.DataConnectorReq{
        DataConnector: dataConnectorbody,
    }
    response, err := client.CreateDataConnector(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
201	数据连接创建成功

错误码

请参见**错误码**。

6.7.2 查询数据连接列表

功能介绍

查询数据连接列表

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/data-connectors

表 6-133 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 6-134 Query 参数

参数	是否必选	参数类型	描述
connector_id	否	String	**参数解释：** 连接ID。获取方法，请参见**获取连接ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
source_type	否	String	**参数解释：** 数据源类别。 **约束限制：** 不涉及 **取值范围：** • RDS_POSTGRES：RDS服务PostgreSQL数据库 • RDS_MYSQL：RDS服务MySQL数据库 • gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及
connector_name	否	String	**参数解释：** 数据连接名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
limit	否	Integer	**参数解释：** 每页返回的资源个数。 **约束限制：** 不涉及 **取值范围：** 1-1000 **默认取值：** 不涉及
offset	否	Integer	**参数解释：** 分页查询起始偏移量。 **约束限制：** 不涉及 **取值范围：** ≥0 **默认取值：** 不涉及
available	否	Boolean	**参数解释：** 数据连接是否有效。 **约束限制：** 不涉及 **取值范围：** true：连接有效 false：连接无效 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-135 响应 Body 参数

参数	参数类型	描述
total_count	Integer	**参数解释：** 数据连接总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
data_connectors	Array of **DataConnectorDetail** objects	**参数解释：** 数据连接详情列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-136 DataConnectorDetail

参数	参数类型	描述
connector_name	String	**参数解释：** 数据连接名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
source_type	String	**参数解释：** 数据连接类型。 **约束限制：** 不涉及 **取值范围：** ● RDS_POSTGRES：RDS服务PostgreSQL数据库 ● RDS_MYSQL：RDS服务MySQL数据库 ● gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及
source_info	String	**参数解释：** 数据源信息，为json格式。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
connector_id	String	**参数解释：** 数据连接ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_time	Long	**参数解释：** 创建时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
last_update_time	Long	**参数解释：** 最后更新时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_by	String	**参数解释：** 创建用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_user	String	**参数解释：** 创建用户名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenant_id	String	**参数解释：** 租户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
last_update_by	String	**参数解释：** 最后更新用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	Integer	**参数解释：** 数据连接状态。 **约束限制：** 不涉及 **取值范围：** ● -1：已删除 ● 0：正常 ● -2：异常 ● 1：使用中 **默认取值：** 不涉及
used_clusters	String	**参数解释：** 使用集群。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
encrypt_type	Integer	**参数解释：** 加密类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

查询数据连接列表成功

```
{
  "total_count" : 1,
  "data_connectors" : [ {
    "connector_id" : "7d169c6c-ab50-4a56-a5d2-240ca20aabda",
    "connector_name" : "test",
    "create_time" : 1681270961,
    "source_type" : "RDS_MYSQL",
    "source_info" : "{\"rds_instance_id\":\"3ce1ae8af8cd43b2bcbd18b7541bb8b0in01\",\"jdbc_url
\":\"jdbc:mysql://192.XXX.XXX.XXX:3306/test?socketTimeout=60000\",\"db_name\":\"test\",\"user_name
\":\"root\",\"driver_path\":\"mrs-public/dirvers/mysql-connector-java-5.1.47.jar\"}",
    "last_update_time" : 1681270961,
    "create_by" : "148c89d603e048b291f5940935b38f46",
    "create_user" : "xxx",
    "tenant_id" : "xxxxe662a7e24cc99bfc858c4558dbf6",
    "last_update_by" : "148c89d603e048b291f5940935b38f46",
    "status" : 0,
    "encrypt_type" : 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.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ListDataConnectorSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
```

```
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDataConnectorRequest request = new ListDataConnectorRequest();
        try {
            ListDataConnectorResponse response = client.listDataConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	查询数据连接列表成功

错误码

请参见**错误码**。

6.7.3 更新数据连接

功能介绍

更新数据连接

调用方法

请参见**如何调用API**。

URI

PUT /v2/{project_id}/data-connectors/{connector_id}

表 6-137 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
connector_id	是	String	**参数解释：** 连接ID。获取方法，请参见**获取连接ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-138 请求 Body 参数

参数	是否必选	参数类型	描述
data_connector	是	**DataConnector** object	**参数解释：** 数据连接。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-139 DataConnector

参数	是否必选	参数类型	描述
connector_name	是	String	**参数解释：** 数据连接名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
source_type	是	String	**参数解释：** 数据连接类型。 **约束限制：** 不涉及 **取值范围：** • RDS_POSTGRES：RDS服务PostgreSQL数据库 • RDS_MYSQL：RDS服务MySQL数据库 • gaussdb-mysql：云数据库GaussDB(for MySQL) **默认取值：** 不涉及

参数	是否必选	参数类型	描述
source_info	是	String	**参数解释：** 数据源信息，为json格式。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 200

表 6-140 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 更新映射请求操作结果。 **约束限制：** 不涉及 **取值范围：** ● succeeded：操作成功 ● failed：操作失败 **默认取值：** 不涉及

请求示例

```
/v2/274ee662a7e24cc99bfc858c4558dbf6/data-connector/7d169c6c-ab50-4a56-a5d2-240ca20aabda

{
  "data_connector" : {
    "connector_name" : "mrs-mysql-connector",
    "source_type" : "RDS_MYSQL",
    "source_info" : "{\"db_name\":\"default\",\"user_name\":\"mrs\",\"password\":\"*****\",\"rds_instance_id
\":\"3aa55f1cb4c3491686936130f21e9f16in01\"}"
  }
}
```

响应示例

状态码： 200

数据连接更新成功

```
{
  "result" : "succeed"
}
```

状态码

状态码	描述
200	数据连接更新成功

错误码

请参见**错误码**。

6.7.4 删除数据连接

功能介绍

删除数据连接

调用方法

请参见**如何调用API**。

URI

DELETE /v2/{project_id}/data-connectors/{connector_id}

表 6-141 路径参数

参数	是否必选	参数类型	描述
connector_id	是	String	**参数解释：** 连接ID。获取方法，请参见**获取连接ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

无

请求示例

无

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class DeleteDataConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteDataConnectorRequest request = new DeleteDataConnectorRequest();
        request.withConnectorId("{connector_id}");
        try {
            DeleteDataConnectorResponse response = client.deleteDataConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteDataConnectorRequest()
        request.connector_id = "{connector_id}"
        response = client.delete_data_connector(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
204	删除数据连接成功

错误码

请参见**错误码**。

6.8 版本元数据查询

6.8.1 展示 MRS 版本列表

功能介绍

展示MRS版本列表

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/metadata/versions

表 6-142 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-143 响应 Body 参数

参数	参数类型	描述
cluster_versions	Array of strings	**参数解释：** 集群版本列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "cluster_versions" : [ "MRS 1.9.2", "MRS 3.1.0", "MRS 3.1.2-LTS.3", "MRS 3.1.5", "MRS 3.2.0-LTS.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.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ShowMrsVersionListSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowMrsVersionListRequest request = new ShowMrsVersionListRequest();
        try {
            ShowMrsVersionListResponse response = client.showMrsVersionList(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

import (
```

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	请求成功

错误码

请参见**错误码**。

6.8.2 查询 MRS 集群版本可用的规格

功能介绍

查询MRS集群版本可用的规格

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/metadata/version/{version_name}/available-flavor

表 6-144 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
version_name	是	String	**参数解释：** 集群版本。例如“MRS 3.1.0”。如果请求客户端不支持自动转义，则需要将空格转义为%20，例如“MRS %203.1.0”。 **约束限制：** 不涉及 **取值范围：** 长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-145 响应 Body 参数

参数	参数类型	描述
version_name	String	**参数解释：** 版本名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
available_flavors	Array of **AzFlavors** objects	**参数解释：** 不同可用区支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-146 AzFlavors

参数	参数类型	描述
az_code	String	**参数解释：** 可用区code。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
az_name	String	**参数解释：** 可用区名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
master	Array of **Flavor** objects	**参数解释：** master节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
core	Array of **Flavor** objects	**参数解释：** core节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task	Array of **Flavor** objects	**参数解释：** task节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 6-147 Flavor

参数	参数类型	描述
flavor_name	String	**参数解释：** 规格名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

获取MRS 3.1.0版本元数据的详细信息

```
/v2/f77c10d14a544393a24e5f0bf53202b6/metadata/version/MRS%203.1.0/available-flavor
```

响应示例

状态码： 200

当前版本可用规格

```
{
  "version_name" : "MRS 3.1.0",
  "available_flavors" : [ {
    "az_code" : "cn-north-7a",
    "az_name" : "可用区1",
    "master" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "core" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "task" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ]
  }, {
    "az_code" : "cn-north-7b",
    "az_name" : "可用区2",
    "master" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "core" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "task" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
```

```
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ]
  }, {
    "az_code" : "cn-north-7c",
    "az_name" : "可用区3",
    "master" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "core" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ],
    "task" : [ {
      "flavor_name" : "Si3.4xlarge.4"
    }, {
      "flavor_name" : "m3.8xlarge.8"
    }, {
      "flavor_name" : "c6.xlarge.4"
    }, {
      "flavor_name" : "m6.xlarge.8"
    } ]
  } ]
}
```


SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowMrsFlavorsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowMrsFlavorsRequest request = new ShowMrsFlavorsRequest();
        request.withVersionName("{version_name}");
        try {
            ShowMrsFlavorsResponse response = client.showMrsFlavors(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowMrsFlavorsRequest()
        request.version_name = "{version_name}"
        response = client.show_mrs_flavors(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	当前版本可用规格

错误码

请参见**错误码**。

6.9 IAM 同步管理接口

6.9.1 获取已经同步的 IAM 用户和用户组

功能介绍

获取已经同步的IAM用户和用户组

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/iam-sync-user

表 6-148 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-149 响应 Body 参数

参数	参数类型	描述
user_names	Array of strings	**参数解释：** 已经同步了的用户。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
group_names	Array of strings	**参数解释：** 已经同步了的用户组。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

已经同步了的用户和用户组

```
{
  "user_names" : [ "user1", "user2" ],
  "group_names" : [ "group1" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v2.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v2.*;
import com.huaweicloud.sdk.mrs.v2.model.*;


public class ShowSyncIamUserSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSyncIamUserRequest request = new ShowSyncIamUserRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowSyncIamUserResponse response = client.showSyncIamUser(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

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

    try:
        request = ShowSyncIamUserRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_sync_iam_user(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	已经同步了的用户和用户组

错误码

请参见**错误码**。

6.9.2 IAM 同步

功能介绍

将IAM用户和用户组同步到manager，指定用户的情况下，会将该用户关联的IAM用户组也同步到manager。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/iam-sync-user

表 6-150 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-151 请求 Body 参数

参数	是否必选	参数类型	描述
is_all_sync	否	Boolean	**参数解释：** 是否是全量同步。 **约束限制：** 不涉及 **取值范围：** • true：全量同步 • false：指定用户、用户组同步 **默认取值：** false
group_names	否	Array of strings	**参数解释：** 指定同步的IAM用户组。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
user_names	否	Array of strings	**参数解释：** 指定同步的IAM用户。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 202

表 6-152 响应 Body 参数

参数	参数类型	描述
state	String	**参数解释：** 请求下发的结果。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

将group1和user1同步到manager。同时会自动将user1关联的IAM用户组也同步到manager

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/iam-sync-user

{
  "is_all_sync" : false,
  "group_names" : [ "groups1" ],
  "user_names" : [ "user1", "user2" ]
}
```

响应示例

状态码： 202

同步请求下发的结果

```
{
  "state" : "synchronizing"
}
```

SDK 代码示例

SDK代码示例如下。

Java

将group1和user1同步到manager。同时会自动将user1关联的IAM用户组也同步到manager

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

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

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

public class UpdateSyncIamUserSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateSyncIamUserRequest request = new UpdateSyncIamUserRequest();
        request.withClusterId("{cluster_id}");
        UpdateSyncRequest body = new UpdateSyncRequest();
        List<String> listbodyUserNames = new ArrayList<>();
        listbodyUserNames.add("user1");
        listbodyUserNames.add("user2");
        List<String> listbodyGroupNames = new ArrayList<>();
        listbodyGroupNames.add("groups1");
        body.withUserNames(listbodyUserNames);
        body.withGroupNames(listbodyGroupNames);
        body.withIsAllSync(false);
        request.withBody(body);
        try {
            UpdateSyncIamUserResponse response = client.updateSyncIamUser(request);
            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

将group1和user1同步到manager。同时会自动将user1关联的IAM用户组也同步到manager

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateSyncIamUserRequest()
        request.cluster_id = "{cluster_id}"
        listUserNamesbody = [
            "user1",
            "user2"
        ]
        listGroupNamesbody = [
            "groups1"
        ]
        request.body = UpdateSyncRequest(
            user_names=listUserNamesbody,
            group_names=listGroupNamesbody,
            is_all_sync=False
        )
        response = client.update_sync_iam_user(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

将group1和user1同步到manager。同时会自动将user1关联的IAM用户组也同步到manager

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateSyncIamUserRequest{}
    request.ClusterId = "{cluster_id}"
    var listUserNamesbody = []string{
        "user1",
        "user2",
    }
    var listGroupNamesbody = []string{
        "groups1",
    }
    isAllSyncUpdateSyncRequest:= false
    request.Body = &model.UpdateSyncRequest{
        UserNames: &listUserNamesbody,
        GroupNames: &listGroupNamesbody,
        IsAllSync: &isAllSyncUpdateSyncRequest,
    }
    response, err := client.UpdateSyncIamUser(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	同步请求下发的结果

错误码

请参见**错误码**。

6.9.3 指定用户、用户组取消同步

功能介绍

指定用户、用户组取消同步

调用方法

请参见**如何调用API**。

URI

DELETE /v2/{project_id}/clusters/{cluster_id}/iam-sync-user

表 6-153 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-154 请求 Body 参数

参数	是否必选	参数类型	描述
group_names	否	Array of strings	**参数解释：** 指定取消同步的IAM用户组。 **约束限制：** 不能超过1000条。 **取值范围：** 不涉及 **默认取值：** 不涉及
user_names	否	Array of strings	**参数解释：** 指定取消同步的IAM用户。 **约束限制：** 不能超过1000条。 **取值范围：** 不涉及 **默认取值：** 不涉及

响应参数

状态码： 202

表 6-155 响应 Body 参数

参数	参数类型	描述
state	String	**参数解释：** 请求下发的结果。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

```
/v2/ff8080828997cb24018a1b2db3440b80/clusters/f7f45c04-4303-411c-9b71-d2cb730dd162/iam-sync-user
```

```
{
  "group_names" : [ "groups1" ],
  "user_names" : [ "user1", "user2" ]
}
```

响应示例

状态码： 202

同步请求下发的结果

```
{
  "state" : "synchronizing"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

public class CancelSyncIamUserSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CancelSyncIamUserRequest request = new CancelSyncIamUserRequest();
        request.withClusterId("{cluster_id}");
        CancelSyncRequest body = new CancelSyncRequest();
        List<String> listbodyUserNames = new ArrayList<>();
        listbodyUserNames.add("user1");
        listbodyUserNames.add("user2");
        List<String> listbodyGroupNames = new ArrayList<>();
        listbodyGroupNames.add("groups1");
        body.withUserNames(listbodyUserNames);
        body.withGroupNames(listbodyGroupNames);
        request.withBody(body);
```

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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CancelSyncIamUserRequest()
        request.cluster_id = "{cluster_id}"
        listUserNamesbody = [
            "user1",
            "user2"
        ]
        listGroupNamesbody = [
            "groups1"
        ]
        request.body = CancelSyncRequest(
            user_names=listUserNamesbody,
            group_names=listGroupNamesbody
        )
        response = client.cancel_sync_iam_user(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CancelSyncIamUserRequest{}
    request.ClusterId = "{cluster_id}"
    var listUserNamesbody = []string{
        "user1",
        "user2",
    }
    var listGroupNamesbody = []string{
        "groups1",
    }
    request.Body = &model.CancelSyncRequest{
        UserNames: &listUserNamesbody,
        GroupNames: &listGroupNamesbody,
    }
    response, err := client.CancelSyncIamUser(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	同步请求下发的结果

错误码

请参见**错误码**。

6.10 标签管理接口

6.10.1 集群操作默认标签

功能介绍

对已有集群启用或关闭集群默认标签。开启后，集群内节点会打上集群默认标签。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/clusters/{cluster_id}/tags/switch

表 6-156 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 6-157 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	**参数解释：** 操作类型，支持创建和删除。 **约束限制：** 不涉及 **取值范围：** ● create：创建集群标签。 ● delete：删除集群标签。 **默认取值：** 不涉及

响应参数

无

请求示例

- 关闭集群默认标签

```
/v2/174ee662a7e24cc99bfc858c455xxxxxx/clusters/848cd341-cbc8-4ac3-82f2-35cd58c077d9/tags/switch

{
  "action" : "delete"
}
```

- 开启集群标签

```
/v2/174ee662a7e24cc99bfc858c455xxxxxx/clusters/848cd341-cbc8-4ac3-82f2-35cd58c077d9/tags/switch

{
  "action" : "create"
}
```

响应示例

状态码： 400

请求失败

```
{
  "error_code" : "MRS.00000000",
  "error_msg" : "标签配额不足"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 关闭集群默认标签

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

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


public class SwitchClusterTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        SwitchClusterTagsRequest request = new SwitchClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        ModifyDefaultTagsRequestBody body = new ModifyDefaultTagsRequestBody();
        body.withAction(ModifyDefaultTagsRequestBody.ActionEnum.fromValue("delete"));
        request.withBody(body);
```

```
        try {
            SwitchClusterTagsResponse response = client.switchClusterTags(request);
            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());
        }
    }
}
```

- 开启集群标签

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

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


public class SwitchClusterTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        SwitchClusterTagsRequest request = new SwitchClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        ModifyDefaultTagsRequestBody body = new ModifyDefaultTagsRequestBody();
        body.withAction(ModifyDefaultTagsRequestBody.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            SwitchClusterTagsResponse response = client.switchClusterTags(request);
            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

- 关闭集群默认标签

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = SwitchClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        request.body = ModifyDefaultTagsRequestBody(
            action="delete"
        )
        response = client.switch_cluster_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 开启集群标签

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

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

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

try:
    request = SwitchClusterTagsRequest()
    request.cluster_id = "{cluster_id}"
    request.body = ModifyDefaultTagsRequestBody(
        action="create"
    )
    response = client.switch_cluster_tags(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- 关闭集群默认标签

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchClusterTagsRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.ModifyDefaultTagsRequestBody{
        Action: model.GetModifyDefaultTagsRequestBodyActionEnum().DELETE,
    }
    response, err := client.SwitchClusterTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 开启集群标签

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchClusterTagsRequest{}
    request.ClusterId = "{cluster_id}"
    request.Body = &model.ModifyDefaultTagsRequestBody{
        Action: model.GetModifyDefaultTagsRequestBodyActionEnum().CREATE,
    }
    response, err := client.SwitchClusterTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
202	请求成功
400	请求失败

错误码

请参见**错误码**。

6.10.2 查询默认标签状态

功能介绍

查询集群默认标签状态

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/tags/status

表 6-158 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-159 响应 Body 参数

参数	参数类型	描述
status	String	**参数解释：** 标签处理状态。 **约束限制：** 不涉及 **取值范围：** ● processing：操作正在处理中。 ● succeed：操作成功。 ● failed：操作失败。 **默认取值：** 不涉及
default_tags_enable	Boolean	**参数解释：** 默认标签是否已开启。 **约束限制：** 不涉及 **取值范围：** ● true：默认标签已开启。 ● false：默认标签没有开启。 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "status" : "succeed",
  "default_tags_enable" : "true"
}
```

状态码： 400

请求失败

```
{
  "error_code" : "MRS.00000000",
  "error_msg" : "请求标签操作错误"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowTagStatusSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTagStatusRequest request = new ShowTagStatusRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowTagStatusResponse response = client.showTagStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowTagStatusRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_tag_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
200	请求成功
400	请求失败

错误码

请参见**错误码**。

6.10.3 查询标签配额

功能介绍

查询标签配额信息

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/clusters/{cluster_id}/tags/quota

表 6-160 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 6-161 Query 参数

参数	是否必选	参数类型	描述
auto_scaling_policy_tags	否	Boolean	**参数解释：** 是否查询弹性伸缩策略标签。 **约束限制：** 不涉及 **取值范围：** true：查询弹性伸缩策略标签 false：不查询弹性伸缩策略标签 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 6-162 响应 Body 参数

参数	参数类型	描述
total_quota	Integer	**参数解释：** 总配额大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
available_quota	Integer	**参数解释：** 可使用配额大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "total_quota" : 10,
  "available_quota" : 2
}
```

状态码： 400

请求失败

```
{
  "error" : {
```

```
  "error_code" : "MRS.00000000",
  "error_msg" : "集群不存在"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowTagQuotaSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTagQuotaRequest request = new ShowTagQuotaRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowTagQuotaResponse response = client.showTagQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v2.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowTagQuotaRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_tag_quota(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

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

更多

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

状态码

状态码	描述
200	请求成功
400	请求失败

错误码

请参见**错误码**。

7 API V1.1

7.1 集群管理接口

7.1.1 创建集群并执行作业

功能介绍

创建一个MRS集群，并在集群中提交一个作业。该接口不兼容Sahara。

（建议优先使用**创建集群V2接口**和**创建集群并提交作业V2接口**来完成创建集群或创建集群并提交作业的功能）

支持同一时间并发创建10个集群。

使用接口前，您需要先获取下的资源信息。

- 通过VPC创建或查询VPC、子网
- 通过ECS创建或查询密钥对
- 通过**终端节点**获取区域信息
- 参考**MRS服务支持的组件**获取MRS版本及对应版本支持的组件信息

接口约束

- 集群登录方式有密码和密钥对两种，两者必选其一。
- 使用密码方式需要配置访问集群节点的root密码，即cluster_master_secret。
- 使用密钥对方式需要配置密钥对名称，即node_public_cert_name。

磁盘参数可以使用volume_type和volume_size表示，也可以使用多磁盘相关的参数（master_data_volume_type、master_data_volume_size、master_data_volume_count、core_data_volume_type、core_data_volume_size和core_data_volume_count）表示，以上两种方式任选一组进行配置。

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/run-job-flow

表 7-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-2 请求 Body 参数

参数	是否必选	参数类型	描述
cluster_version	是	String	**参数解释：** 集群版本。例如：MRS 3.1.0。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_name	是	String	**参数解释：** 集群名称，不允许相同。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
master_node_num	否	Integer	**参数解释：** Master节点数量。 **约束限制：** 启用集群高可用功能时配置为2，不启用集群高可用功能时配置为1。MRS 3.x版本暂时不支持该参数配置为1。 **取值范围：** 不涉及 **默认取值：** 不涉及
core_node_num	否	Integer	**参数解释：** Core节点数量。Core节点默认的最大值为500，如果用户需要的Core节点数大于500，请申请扩大配额。 **约束限制：** 不涉及 **取值范围：** 1-500 **默认取值：** 不涉及
billing_type	是	Integer	**参数解释：** 集群的计费模式。 **约束限制：** 不涉及 **取值范围：** 12：表示按需计费。接口调用仅支持创建按需计费集群。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_center	是	String	**参数解释：** 集群区域信息，请参见**终端节点及区域**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpc	是	String	**参数解释：** 子网所在VPC名称。通过VPC管理控制台获取名称。 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 在“虚拟私有云”页面的列表中即可获取VPC名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
master_node_size	否	String	**参数解释：** Master节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。当前支持主机规格的配型由CPU+内存+Disk共同决定。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。 该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
core_node_size	否	String	**参数解释：** Core节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。 该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
component_list	是	Array of **ComponentAmbV11** objects	**参数解释：** 服务组件安装列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
available_zone_id	是	String	**参数解释：** 可用分区ID。以下仅包含部分可用区ID。更多局点可通过**查询可用区信息**接口来获取各可用分区的ID。 **约束限制：** 不涉及 **取值范围：** • 华北-北京一可用区1（cn-north-1a）：ae04cf9d61544df3806a3feeb401b204 • 华北-北京一可用区2（cn-north-1b）：d573142f24894ef3bd3664de068b44b0 • 华东-上海二可用区1（cn-east-2a）：72d50cedc49846b9b42c21495f38d81c • 华东-上海二可用区2（cn-east-2b）：38b0f7a602344246bcb0da47b5d548e7 • 华东-上海二可用区3（cn-east-2c）：5547fd6bf8f84bb5a7f9db062ad3d015 • 华南-广州可用区1（cn-south-1a）：34f5ff4865cf4ed6b270f15382ebdec5 • 华南-广州可用区2（cn-south-2b）：043c7e39ecb347a08dc8fcb6c35a274e • 华南-广州可用区3（cn-south-1c）：af1687643e8c4ec1b34b688e4e3b8901 • 华北-北京四可用区1（cn-north-4a）：effdcbc7d4d64a02aa1fa26b42f56533 • 华北-北京四可用区2（cn-north-4b）：

参数	是否必选	参数类型	描述
			a0865121f83b41cbafce65930a22a6e8 ● 华北-北京四可用区3（cn-north-4c）：2dcb154ac2724a6d92e9bcc859657c1e **默认取值：** 不涉及
vpc_id	是	String	**参数解释：** 子网所在VPC ID。通过VPC管理控制台获取ID。 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 在“虚拟私有云”页面的列表中即可获取VPC ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_id	是	String	**参数解释：** 子网ID。通过VPC管理控制台获取子网ID。 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网。 4. 单击对应子网名称，获取“网络ID”。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
subnet_name	是	String	**参数解释：** 子网名称。 通过VPC管理控制台获取子网名称： 1. 登录管理控制台。 2. 单击“虚拟私有云”，从左侧列表选择虚拟私有云。 3. 单击对应虚拟私有云所在行的“子网个数”查看子网，获取子网名称。 **约束限制：** “subnet_id”和“subnet_name”必须至少填写一个，当这两个参数同时配置但是不匹配同一个子网时，集群会创建失败，请仔细填写参数。当仅填写“subnet_name”一个参数且VPC下存在同名子网时，创建集群时以VPC平台第一个名称的子网为准。推荐使用“subnet_id”。 **取值范围：** 不涉及 **默认取值：** 不涉及
security_groups_id	否	String	**参数解释：** 集群安全组的ID。 ● 当该ID为空时MRS后台会自己创建安全组，自动创建的安全组名称以mrs_{cluster_name}开头。 ● 当该ID不为空时，表示使用固定安全组来创建集群，传入的ID必须是当前租户中包含的安全组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
add_jobs	否	Array of **AddJobsReqV11** objects	**参数解释：** 创建集群时可同时提交作业，当前版本暂时只支持新增一个作业。 **约束限制：** 不能超过1条。 **取值范围：** 不涉及 **默认取值：** 不涉及
volume_size	否	Integer	**参数解释：** Master和Core节点数据磁盘存储空间，单位为GB。为增大数据存储容量，创建集群时可同时添加磁盘。可以根据如下应用场景合理选择磁盘存储空间大小： • 数据存储和计算分离，数据存储在OBS系统中，集群费用相对较低，计算性能不高，并且集群随时可以删除，建议数据计算不频繁场景下使用。 • 数据存储和计算不分离，数据存储在HDFS中，集群费用相对较高，计算性能高，集群需要长期存在，建议数据计算频繁场景下使用。 **约束限制：** 不建议使用该参数，详情请参考volume_type参数的说明。 **取值范围：** 100-32000 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
volume_type	否	String	**参数解释：** Master和Core节点的磁盘存储类别，目前支持SATA、SAS、SSD和GPSSD。磁盘参数可以使用volume_type和volume_size表示，也可以使用多磁盘相关的参数表示。volume_type和volume_size这两个参数如果与多磁盘参数同时出现，系统优先读取volume_type和volume_size参数。建议使用多磁盘参数。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及
master_data_volume_type	否	String	**参数解释：** 该参数为多磁盘参数，表示Master节点数据磁盘存储类别，目前支持SATA、SAS、SSD和GPSSD。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及

参数	是否必选	参数类型	描述
master_data_volume_size	否	Integer	**参数解释：** 该参数为多磁盘参数，表示Master节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。传值只需填数字，不需要带单位GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及
master_data_volume_count	否	Integer	**参数解释：** 该参数为多磁盘参数，表示Master节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 只能为1。 **默认取值：** 1
core_data_volume_type	否	String	**参数解释：** 该参数为多磁盘参数，表示Core节点数据磁盘存储类别，目前支持SATA、SAS、SSD和GPSSD。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及

参数	是否必选	参数类型	描述
core_data_volume_size	否	Integer	**参数解释：** 该参数为多磁盘参数，表示Core节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。传值只需填数字，不需要带单位GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及
core_data_volume_count	否	Integer	**参数解释：** 该参数为多磁盘参数，表示Core节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 1-20 **默认取值：** 不涉及
task_node_groups	否	Array of **TaskNodeGroup** objects	**参数解释：** Task节点列表信息。 **约束限制：** 不能超过1条。 **取值范围：** 不涉及 **默认取值：** 不涉及
bootstrap_scripts	否	Array of **BootstrapScript** objects	**参数解释：** 配置引导操作脚本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_public_cert_name	否	String	**参数解释：** 密钥对名称。用户可以使用密钥对方式登录集群节点。 **约束限制：** 当“login_mode”配置为“1”时，请求消息体中包含node_public_cert_name字段。 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_admin_secret	否	String	**参数解释：** 配置MRS Manager管理员用户的密码。 **约束限制：** 不涉及 **取值范围：** ● 密码长度应在8～26个字符之间 ● 不能与用户名或者倒序用户名相同 ● 必须包含如下4种字符的组合 – 至少一个小写字母 – 至少一个大写字母 – 至少一个数字 – 至少一个特殊字符：!@$%^-_=+[{}]:,./? **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_master_secret	否	String	**参数解释：** 配置访问集群节点的root密码。 **约束限制：** 当“login_mode”配置为“0”时，请求消息体中包含cluster_master_secret字段。 **取值范围：** 密码设置约束如下： ● 字符串类型，可输入的字符串长度为8-26。 ● 至少包含4种字符组合，如大写字母，小写字母，数字，特殊字符（!@$%^-_=+[{}]:,./?），但不能包含空格。 ● 不能与用户名或者倒序用户名相同。 **默认取值：** 不涉及
safe_mode	是	Integer	**参数解释：** MRS集群运行模式。 **约束限制：** 不涉及 **取值范围：** ● 0：普通集群，表示Kerberos认证关闭，用户可使用集群提供的所有功能。 ● 1：安全集群，表示Kerberos认证开启，普通用户无权限使用MRS集群的“文件管理”和“作业管理”功能，并且无法查看Hadoop、Spark的作业记录以及集群资源使用情况。如果需要使用集群更多功能，需要找MRS Manager的管理员分配权限。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cluster_type	否	Integer	**参数解释：** 集群类型。暂不支持通过接口方式创建混合集群。 **约束限制：** 不涉及 **取值范围：** ● 0：分析集群。 ● 1：流式集群。 **默认取值：** 0
log_collection	否	Integer	**参数解释：** 集群创建失败时，是否收集失败日志。默认设置为1，将创建OBS桶仅用于MRS集群创建失败时的日志收集。 **约束限制：** 不涉及 **取值范围：** ● 0：不收集 ● 1：收集 **默认取值：** 1
enterprise_project_id	否	String	**参数解释：** 企业项目ID。创建集群时，给集群绑定企业项目ID。默认设置为0，表示为default企业项目。获取方式请参见《企业管理API参考》的“查询企业项目列表”响应消息表“enterprise_project字段数据结构说明”的“id”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 0

参数	是否必选	参数类型	描述
tags	否	Array of **Tag** objects	**参数解释：** 集群的标签信息。 **约束限制：** 同一个集群最多能使用20个tag，tag的名称（key）不能重复标签的键/值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **取值范围：** 不涉及 **默认取值：** 不涉及
login_mode	否	Integer	**参数解释：** 集群登录方式。 **约束限制：** ● 当“login_mode”配置为“0”时，请求消息体中包含cluster_master_secret字段。 ● 当“login_mode”配置为“1”时，请求消息体中包含node_public_cert_name字段。 **取值范围：** ● 0：密码方式 ● 1：密钥对方式 **默认取值：** 1

参数	是否必选	参数类型	描述
node_groups	否	Array of NodeGroupV11 objects	**参数解释：** 节点列表信息。 **约束限制：** 如下参数和该参数任选一组进行配置即可。 master_node_num、master_node_size、core_node_num、core_node_size、master_data_volume_type、master_data_volume_size、master_data_volume_count、core_data_volume_type、core_data_volume_size、core_data_volume_count、volume_type、volume_size、task_node_groups。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-3 ComponentAmbV11

参数	是否必选	参数类型	描述
component_name	是	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 7-4 AddJobsReqV11

参数	是否必选	参数类型	描述
job_type	是	Integer	**参数解释：** 作业类型码。 **约束限制：** 不涉及 **取值范围：** ● 1：MapReduce ● 2：Spark ● 3：Hive Script ● 4：HiveSQL（当前不支持） ● 5：DistCp，导入、导出数据，（当前不支持）。 ● 6：Spark Script ● 7：Spark SQL，提交SQL语句，（当前不支持）。 **默认取值：** 不涉及
job_name	是	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
jar_path	否	String	**参数解释：** 执行程序Jar包或sql文件地址。 **约束限制：** 不涉及 **取值范围：** 需要满足如下要求： ● 最多为1023字符，不能包含;|&>,<'$特殊字符，且不可为空或全空格。 ● 文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 – OBS：以“s3a://”开头。不支持KMS加密的文件或程序。 – HDFS：以“/”开头。 ● Spark Script需要以“.sql”结尾，MapReduce和Spark Jar需要以“.jar”结尾，sql和jar不区分大小写。 **默认取值：** 不涉及
arguments	否	String	**参数解释：** 程序执行的关键参数，该参数由用户程序内的函数指定，MRS只负责参数的传入。 **约束限制：** 不涉及 **取值范围：** 最多为150000字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
input	否	String	**参数解释：** 数据输入地址。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 ● OBS：以“s3a://”开头。不支持KMS加密的文件或程序。 ● HDFS：以“/”开头。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
output	否	String	**参数解释：** 数据输出地址。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 ● OBS：以“s3a://”开头。 ● HDFS：以“/”开头。 如果该路径不存在，系统会自动创建。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_log	否	String	**参数解释：** 作业日志存储地址，该日志信息记录作业运行状态。文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 ● OBS：以“s3a://”开头。 ● HDFS：以“/”开头。 **约束限制：** 不涉及 **取值范围：** 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
hive_script_path	否	String	**参数解释：** sql程序路径，仅Spark Script和Hive Script作业需要使用此参数。 **约束限制：** 不涉及 **取值范围：** 需要满足如下要求： ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 文件可存储于HDFS或者OBS中，不同的文件系统对应的路径存在差异。 – OBS：以“s3a://”开头。不支持KMS加密的文件或程序。 – HDFS：以“/”开头。 ● 需要以“.sql”结尾，sql不区分大小写。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
hql	否	String	**参数解释：** HQL脚本语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
shutdown_cluster	否	Boolean	**参数解释：** 作业执行完成后，是否删除集群。 **约束限制：** 不涉及 **取值范围：** ● true：作业执行完成后，删除集群。 ● false：作业执行完成后，不删除集群。 **默认取值：** 不涉及
submit_job_once_cluster_run	是	Boolean	**参数解释：** 创建集群时是否同时提交作业。此处应设置为true。 **约束限制：** 不涉及 **取值范围：** ● true：创建集群同时提交作业 ● false：单独提交作业 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
file_action	否	String	**参数解释：** 数据导入导出。 **约束限制：** 不涉及 **取值范围：** ● import：数据的导入操作。 ● export：数据的导出操作。 **默认取值：** 不涉及

表 7-5 TaskNodeGroup

参数	是否必选	参数类型	描述
node_num	是	Integer	**参数解释：** Task节点数量。 **约束限制：** Core与Task节点总数最大为500个。 **取值范围：** 0-500 **默认取值：** 不涉及
node_size	是	String	**参数解释：** Task节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_volume_type	是	String	**参数解释：** Task节点数据磁盘存储类别，目前支持SATA、SAS和SSD等。 **约束限制：** 不涉及 **取值范围：** • SATA：普通IO • SAS：高IO • SSD：超高IO • GPSSD：通用型SSD **默认取值：** 不涉及
data_volume_count	是	Integer	**参数解释：** Task节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 0-20 **默认取值：** 不涉及
data_volume_size	是	Integer	**参数解释：** Task节点数据磁盘存储大小。传值只需填数字，不需要带单位GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及
auto_scaling_policy	否	**AutoScalingPolicy** object	**参数解释：** 弹性伸缩规则信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-6 BootstrapScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 引导操作脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的引导操作脚本名称不允许相同。不能以空格开头，只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
uri	是	String	**参数解释：** 引导操作脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。例如输入MRS提供的公共样例脚本路径。示例：s3a://bootstrap/presto/presto-install.sh，其中安装dualroles时，presto-install.sh脚本参数为dualroles,安装worker时，presto-install.sh脚本参数为worker。根据Presto使用习惯，建议您在Active Master节点上安装dualroles，在Core节点上安装worker。 ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
parameters	否	String	**参数解释：** 引导操作脚本参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
nodes	是	Array of strings	**参数解释：** 引导操作脚本所执行的节点类型，包含master、core和task三种类型。 **约束限制：** 节点类型必须为小写字母。 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 引导操作脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本只运行在主Master节点上。 ● false：引导操作脚本可运行在所有Master节点上。 **默认取值：** false

参数	是否必选	参数类型	描述
fail_action	是	String	**参数解释：** 引导操作脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此引导操作是否执行成功，则集群都能继续安装和启动。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue
before_component_start	否	Boolean	**参数解释：** 引导操作脚本执行的时间。目前支持“组件启动前”和“组件启动后”两种类型。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本在组件启动前执行。 ● false：引导操作脚本在组件启动后执行。 **默认取值：** false
start_time	否	Long	**参数解释：** 单个引导操作脚本的执行时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
state	否	String	**参数解释：** 单个引导操作脚本的运行状态。 **约束限制：** 不涉及 **取值范围：** ● PENDING：挂起 ● IN_PROGRESS：正在处理 ● SUCCESS：处理成功 ● FAILURE：处理失败 **默认取值：** 不涉及
action_stages	否	Array of strings	**参数解释：** 选择引导操作脚本执行的时间。 ● BEFORE_COMPONENT_FIRST_START: 组件首次启动后 ● AFTER_COMPONENT_FIRST_START: 组件首次启动前 ● BEFORE_SCALE_IN: 缩容前 ● AFTER_SCALE_IN: 缩容后 ● BEFORE_SCALE_OUT: 扩容前 ● AFTER_SCALE_OUT: 扩容后 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-7 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 最大长度128个字符，不能为空字符串。 ● 同一资源的key值不能重复。 ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 最大长度255个字符，可以为空字符串。 ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

表 7-8 NodeGroupV11

参数	是否必选	参数类型	描述
group_name	是	String	**参数解释：** 节点组名。 **约束限制：** 不涉及 **取值范围：** ● master_node_default_group ● core_node_analysis_group ● core_node_streaming_group ● task_node_analysis_group ● task_node_streaming_group **默认取值：** 不涉及
node_num	是	Integer	**参数解释：** 节点数量。 **约束限制：** Core与Task节点总数最大为500个。 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_size	是	String	**参数解释：** 节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。MRS当前支持主机规格的配型由CPU+内存+Disk共同决定。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume_size	否	String	**参数解释：** 节点系统磁盘存储大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume_type	否	String	**参数解释：** 节点系统磁盘存储类别，目前支持SATA、SAS和SSD等。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_volume_type	否	String	**参数解释：** 节点数据磁盘存储类别，目前支持SATA、SAS和SSD等。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及
data_volume_count	否	Integer	**参数解释：** 节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 0-20 **默认取值：** 不涉及
data_volume_size	否	Integer	**参数解释：** 节点数据磁盘存储大小。单位为GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
auto_scaling_policy	否	AutoScalingPolicy object	**参数解释：** 弹性伸缩规则信息。 **约束限制：** 当“group_name”配置为“task_node_analysis_group”或“task_node_streaming_group”时该参数有效。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-9 AutoScalingPolicy

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** • true：开启自动伸缩规则。 • false：不开启自动伸缩规则。 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及
exec_scripts	否	Array of **ScaleScript** objects	**参数解释：** 弹性伸缩自定义自动化脚本列表。若该参数为空表示不启用自动化脚本。 **约束限制：** 不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-10 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
start_time	是	String	**参数解释：** 资源计划的起始时间。格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

表 7-11 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。在一个节点组范围内，不允许重名。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度小于等于1024个字符。 **默认取值：** 不涉及
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** ● scale_out：扩容 ● scale_in：缩容 **默认取值：** 不涉及
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-12 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见**"弹性伸缩指标列表"**。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** ● LT：小于 ● GT：大于 ● LTOE：小于等于 ● GTOE：大于等于 **默认取值：** 不涉及
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-200 **默认取值：** 不涉及

表 7-13 ScaleScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩自定义自动化脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的自定义自动化脚本名称不允许相同。只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 自定义自动化脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。示例：s3a://XXX/scale.sh ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 自定义自动化脚本参数。多个参数间用空格隔开。可以传入以下系统预定义参数： ● ${mrs_scale_node_num}：扩缩容节点数 ● ${mrs_scale_type}：扩缩容类型，扩容为scale_out，缩容为scale_in ● ${mrs_scale_node_hostnames}：扩缩容的节点主机名称 ● ${mrs_scale_node_ips}：扩缩容的节点IP ● ${mrs_scale_rule_name}：触发扩缩容的规则名 其他用户自定义参数使用方式与普通shell脚本相同，多个参数中间用空格隔开。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 自定义自动化脚本所执行的节点组名称（非自定义集群也可使用节点类型，包含Master、Core和Task三种类型）。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 自定义自动化脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：自定义自动化脚本只运行在主Master节点上。 ● false：自定义自动化脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	是	String	**参数解释：** 自定义自动化脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此自定义自动化脚本是否执行成功，则集群都能继续安装和启动。由于缩容成功无法回滚，因此缩容后执行的脚本“fail_action”必须设置为“continue”。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
action_stage	是	String	**参数解释：** 脚本执行时机。 **约束限制：** 不涉及 **取值范围：** ● before_scale_out：扩容前 ● before_scale_in：缩容前 ● after_scale_out：扩容后 ● after_scale_in：缩容后 **默认取值：** 不涉及

响应参数

状态码： 200

表 7-14 响应 Body 参数

参数	参数类型	描述
result	Boolean	**参数解释：** 操作结果。 **约束限制：** 不涉及 **取值范围：** ● true：操作成功 ● false：操作失败 **默认取值：** 不涉及
msg	String	**参数解释：** 系统提示信息，可为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
cluster_id	String	**参数解释：** 集群创建成功后系统返回的集群ID值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- 使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
POST https://{endpoint}/v1.1/{project_id}/run-job-flow

{
  "billing_type" : 12,
  "data_center" : "",
  "available_zone_id" : "d573142f24894ef3bd3664de068b44b0",
  "cluster_name" : "mrs_HEbK",
  "cluster_version" : "MRS 3.1.0",
  "safe_mode" : 0,
  "cluster_type" : 0,
  "component_list" : [ {
    "component_name" : "Hadoop"
  }, {
    "component_name" : "Spark"
  }, {
    "component_name" : "HBase"
  }, {
    "component_name" : "Hive"
  }, {
    "component_name" : "Presto"
  }, {
    "component_name" : "Tez"
  }, {
    "component_name" : "Hue"
  }, {
    "component_name" : "Loader"
  }, {
    "component_name" : "Flink"
  } ],
  "vpc" : "vpc-4b1c",
  "vpc_id" : "4a365717-67be-4f33-80c5-98e98a813af8",
  "subnet_id" : "67984709-e15e-4e86-9886-d76712d4e00a",
  "subnet_name" : "subnet-4b44",
  "security_groups_id" : "4820eace-66ad-4f2c-8d46-cf340e3029dd",
  "enterprise_project_id" : "0",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 2,
```

```
  "node_size" : "s3.xlarge.2.linux.bigdata",
  "root_volume_size" : 480,
  "root_volume_type" : "SATA",
  "data_volume_type" : "SATA",
  "data_volume_count" : 1,
  "data_volume_size" : 600
}, {
  "group_name" : "core_node_analysis_group",
  "node_num" : 3,
  "node_size" : "s3.xlarge.2.linux.bigdata",
  "root_volume_size" : 480,
  "root_volume_type" : "SATA",
  "data_volume_type" : "SATA",
  "data_volume_count" : 1,
  "data_volume_size" : 600
}, {
  "group_name" : "task_node_analysis_group",
  "node_num" : 2,
  "node_size" : "s3.xlarge.2.linux.bigdata",
  "root_volume_size" : 480,
  "root_volume_type" : "SATA",
  "data_volume_type" : "SATA",
  "data_volume_count" : 0,
  "data_volume_size" : 600,
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 1,
    "max_capacity" : "3",
    "resources_plans" : [ {
      "period_type" : "daily",
      "start_time" : "9:50",
      "end_time" : "10:20",
      "min_capacity" : 2,
      "max_capacity" : 3
    }, {
      "period_type" : "daily",
      "start_time" : "10:20",
      "end_time" : "12:30",
      "min_capacity" : 0,
      "max_capacity" : 2
    } ],
    "exec_scripts" : [ {
      "name" : "before_scale_out",
      "uri" : "s3a://XXX/zeppelin_install.sh",
      "parameters" : "${mrs_scale_node_num} ${mrs_scale_type} xxx",
      "nodes" : [ "master", "core", "task" ],
      "active_master" : "true",
      "action_stage" : "before_scale_out",
      "fail_action" : "continue"
    }, {
      "name" : "after_scale_out",
      "uri" : "s3a://XXX/storm_rebalance.sh",
      "parameters" : "${mrs_scale_node_hostnames} ${mrs_scale_node_ips}",
      "nodes" : [ "master", "core", "task" ],
      "active_master" : "true",
      "action_stage" : "after_scale_out",
      "fail_action" : "continue"
    } ],
    "rules" : [ {
      "name" : "default-expand-1",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNMemoryAvailablePercentage",
        "metric_value" : "25",
        "comparison_operator" : "LT",
        "evaluation_periods" : 10
      }
```

```
    }, {
      "name" : "default-shrink-1",
      "adjustment_type" : "scale_in",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNMemoryAvailablePercentage",
        "metric_value" : "70",
        "comparison_operator" : "GT",
        "evaluation_periods" : 10
      }
    } ]
  }
} ],
"login_mode" : 1,
"cluster_master_secret" : "",
"cluster_admin_secret" : "",
"log_collection" : 1,
"add_jobs" : [ {
  "job_type" : 1,
  "job_name" : "tenji111",
  "jar_path" : "s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
  "arguments" : "wordcount",
  "input" : "s3a://bigdata/input/wd_1k/",
  "output" : "s3a://bigdata/ouput/",
  "job_log" : "s3a://bigdata/log/",
  "shutdown_cluster" : true,
  "file_action" : "",
  "submit_job_once_cluster_run" : true,
  "hql" : "",
  "hive_script_path" : ""
} ],
"bootstrap_scripts" : [ {
  "name" : "Modify os config",
  "uri" : "s3a://XXX/modify_os_config.sh",
  "parameters" : "param1 param2",
  "nodes" : [ "master", "core", "task" ],
  "active_master" : "false",
  "before_component_start" : "true",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "BEFORE_COMPONENT_FIRST_START", "BEFORE_SCALE_IN" ]
}, {
  "name" : "Install zepplin",
  "uri" : "s3a://XXX/zeppelin_install.sh",
  "parameters" : "",
  "nodes" : [ "master" ],
  "active_master" : "true",
  "before_component_start" : "false",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
} ]
}
```

- 不使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
POST https://{endpoint}/v1.1/{project_id}/run-job-flow

{
  "billing_type" : 12,
  "data_center" : "",
  "master_node_num" : 2,
  "master_node_size" : "s3.2xlarge.2.linux.bigdata",
  "core_node_num" : 3,
  "core_node_size" : "s1.xlarge.linux.bigdata",
  "available_zone_id" : "d573142f24894ef3bd3664de068b44b0",
```



```
"cluster_name" : "newcluster",
"vpc" : "vpc1",
"vpc_id" : "5b7db34d-3534-4a6e-ac94-023cd36aaf74",
"subnet_id" : "815bece0-fd22-4b65-8a6e-15788c99ee43",
"subnet_name" : "subnet",
"security_groups_id" : "845bece1-fd22-4b45-7a6e-14338c99ee43",
"tags" : [ {
  "key" : "key1",
  "value" : "value1"
}, {
  "key" : "key2",
  "value" : "value2"
} ],
"cluster_version" : "MRS 3.1.0",
"cluster_type" : 0,
"master_data_volume_type" : "SATA",
"master_data_volume_size" : 600,
"master_data_volume_count" : 1,
"core_data_volume_type" : "SATA",
"core_data_volume_size" : 600,
"core_data_volume_count" : 2,
"node_public_cert_name" : "SSHkey-bba1",
"safe_mode" : 0,
"log_collection" : 1,
"task_node_groups" : [ {
  "node_num" : 2,
  "node_size" : "s3.xlarge.2.linux.bigdata",
  "data_volume_type" : "SATA",
  "data_volume_count" : 1,
  "data_volume_size" : 600,
  "auto_scaling_policy" : {
    "auto_scaling_enable" : true,
    "min_capacity" : 1,
    "max_capacity" : "3",
    "resources_plans" : [ {
      "period_type" : "daily",
      "start_time" : "9: 50",
      "end_time" : "10: 20",
      "min_capacity" : 2,
      "max_capacity" : 3
    }, {
      "period_type" : "daily",
      "start_time" : "10: 20",
      "end_time" : "12: 30",
      "min_capacity" : 0,
      "max_capacity" : 2
    } ],
    "exec_scripts" : [ {
      "name" : "before_scale_out",
      "uri" : "s3a: //XXX/zeppelin_install.sh",
      "parameters" : "${mrs_scale_node_num}${mrs_scale_type}xxx",
      "nodes" : [ "master", "core", "task" ],
      "active_master" : "true",
      "action_stage" : "before_scale_out",
      "fail_action" : "continue"
    }, {
      "name" : "after_scale_out",
      "uri" : "s3a: //XXX/storm_rebalance.sh",
      "parameters" : "${mrs_scale_node_hostnames}${mrs_scale_node_ips}",
      "nodes" : [ "master", "core", "task" ],
      "active_master" : "true",
      "action_stage" : "after_scale_out",
      "fail_action" : "continue"
    } ],
    "rules" : [ {
      "name" : "default-expand-1",
      "adjustment_type" : "scale_out",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : 1,
```

```
      "trigger" : {
        "metric_name" : "YARNMemoryAvailablePercentage",
        "metric_value" : "25",
        "comparison_operator" : "LT",
        "evaluation_periods" : 10
      }
    }, {
      "name" : "default-shrink-1",
      "adjustment_type" : "scale_in",
      "cool_down_minutes" : 5,
      "scaling_adjustment" : 1,
      "trigger" : {
        "metric_name" : "YARNMemoryAvailablePercentage",
        "metric_value" : "70",
        "comparison_operator" : "GT",
        "evaluation_periods" : 10
      }
    } ]
  }
} ],
"component_list" : [ {
  "component_name" : "Hadoop"
}, {
  "component_name" : "Spark"
}, {
  "component_name" : "HBase"
}, {
  "component_name" : "Hive"
} ],
"add_jobs" : [ {
  "job_type" : 1,
  "job_name" : "tenji111",
  "jar_path" : "s3a: //bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
  "arguments" : "wordcount",
  "input" : "s3a: //bigdata/input/wd_1k/",
  "output" : "s3a: //bigdata/ouput/",
  "job_log" : "s3a: //bigdata/log/",
  "shutdown_cluster" : true,
  "file_action" : "",
  "submit_job_once_cluster_run" : true,
  "hql" : "",
  "hive_script_path" : ""
} ],
"bootstrap_scripts" : [ {
  "name" : "Modifyosconfig",
  "uri" : "s3a: //XXX/modify_os_config.sh",
  "parameters" : "param1param2",
  "nodes" : [ "master", "core", "task" ],
  "active_master" : "false",
  "before_component_start" : "true",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "BEFORE_COMPONENT_FIRST_START", "BEFORE_SCALE_IN" ]
}, {
  "name" : "Installzepplin",
  "uri" : "s3a: //XXX/zeppelin_install.sh",
  "parameters" : "",
  "nodes" : [ "master" ],
  "active_master" : "true",
  "before_component_start" : "false",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
} ]
}
```

- 使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。


```
POST https://{endpoint}/v1.1/{project_id}/run-job-flow

{
  "billing_type" : 12,
  "data_center" : "",
  "available_zone_id" : "d573142f24894ef3bd3664de068b44b0",
  "cluster_name" : "mrs_HEbK",
  "cluster_version" : "MRS 3.1.0",
  "safe_mode" : 0,
  "cluster_type" : 0,
  "component_list" : [ {
    "component_name" : "Hadoop"
  }, {
    "component_name" : "Spark"
  }, {
    "component_name" : "HBase"
  }, {
    "component_name" : "Hive"
  }, {
    "component_name" : "Presto"
  }, {
    "component_name" : "Tez"
  }, {
    "component_name" : "Hue"
  }, {
    "component_name" : "Loader"
  }, {
    "component_name" : "Flink"
  } ],
  "vpc" : "vpc-4b1c",
  "vpc_id" : "4a365717-67be-4f33-80c5-98e98a813af8",
  "subnet_id" : "67984709-e15e-4e86-9886-d76712d4e00a",
  "subnet_name" : "subnet-4b44",
  "security_groups_id" : "4820eace-66ad-4f2c-8d46-cf340e3029dd",
  "enterprise_project_id" : "0",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ],
  "node_groups" : [ {
    "group_name" : "master_node_default_group",
    "node_num" : 1,
    "node_size" : "s3.xlarge.2.linux.bigdata",
    "root_volume_size" : 480,
    "root_volume_type" : "SATA",
    "data_volume_type" : "SATA",
    "data_volume_count" : 1,
    "data_volume_size" : 600
  }, {
    "group_name" : "core_node_analysis_group",
    "node_num" : 1,
    "node_size" : "s3.xlarge.2.linux.bigdata",
    "root_volume_size" : 480,
    "root_volume_type" : "SATA",
    "data_volume_type" : "SATA",
    "data_volume_count" : 1,
    "data_volume_size" : 600
  } ],
  "login_mode" : 1,
  "cluster_master_secret" : "",
  "cluster_admin_secret" : "",
  "log_collection" : 1,
  "add_jobs" : [ {
    "job_type" : 1,
    "job_name" : "tenji111",
    "jar_path" : "s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
```

```
  "arguments" : "wordcount",
  "input" : "s3a://bigdata/input/wd_1k/",
  "output" : "s3a://bigdata/ouput/",
  "job_log" : "s3a://bigdata/log/",
  "shutdown_cluster" : true,
  "file_action" : "",
  "submit_job_once_cluster_run" : true,
  "hql" : "",
  "hive_script_path" : ""
 } ],
 "bootstrap_scripts" : [ {
  "name" : "Modify os config",
  "uri" : "s3a://XXX/modify_os_config.sh",
  "parameters" : "param1 param2",
  "nodes" : [ "master", "core", "task" ],
  "active_master" : "false",
  "before_component_start" : "true",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "BEFORE_COMPONENT_FIRST_START", "BEFORE_SCALE_IN" ]
 }, {
  "name" : "Install zepplin",
  "uri" : "s3a://XXX/zeppelin_install.sh",
  "parameters" : "",
  "nodes" : [ "master" ],
  "active_master" : "true",
  "before_component_start" : "false",
  "start_time" : "1667892101",
  "state" : "IN_PROGRESS",
  "fail_action" : "continue",
  "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
 } ]
}
```

- 不使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

```
POST https://{endpoint}/v1.1/{project_id}/run-job-flow

{
 "billing_type" : 12,
 "data_center" : "",
 "master_node_num" : 1,
 "master_node_size" : "s3.2xlarge.2.linux.bigdata",
 "core_node_num" : 1,
 "core_node_size" : "s1.xlarge.linux.bigdata",
 "available_zone_id" : "d573142f24894ef3bd3664de068b44b0",
 "cluster_name" : "newcluster",
 "vpc" : "vpc1",
 "vpc_id" : "5b7db34d-3534-4a6e-ac94-023cd36aaf74",
 "subnet_id" : "815bece0-fd22-4b65-8a6e-15788c99ee43",
 "subnet_name" : "subnet",
 "security_groups_id" : "",
 "enterprise_project_id" : "0",
 "tags" : [ {
  "key" : "key1",
  "value" : "value1"
 }, {
  "key" : "key2",
  "value" : "value2"
 } ],
 "cluster_version" : "MRS 3.1.0",
 "cluster_type" : 0,
 "master_data_volume_type" : "SATA",
 "master_data_volume_size" : 600,
 "master_data_volume_count" : 1,
 "core_data_volume_type" : "SATA",
 "core_data_volume_size" : 600,
 "core_data_volume_count" : 1,
```



```
  "login_mode" : 1,
  "node_public_cert_name" : "SSHkey-bba1",
  "safe_mode" : 0,
  "cluster_admin_secret" : "******",
  "log_collection" : 1,
  "component_list" : [ {
    "component_name" : "Hadoop"
  }, {
    "component_name" : "Spark"
  }, {
    "component_name" : "HBase"
  }, {
    "component_name" : "Hive"
  }, {
    "component_name" : "Presto"
  }, {
    "component_name" : "Tez"
  }, {
    "component_name" : "Hue"
  }, {
    "component_name" : "Loader"
  }, {
    "component_name" : "Flink"
  } ],
  "add_jobs" : [ {
    "job_type" : 1,
    "job_name" : "tenji111",
    "jar_path" : "s3a://bigdata/program/hadoop-mapreduce-examples-XXX.jar",
    "arguments" : "wordcount",
    "input" : "s3a://bigdata/input/wd_1k/",
    "output" : "s3a://bigdata/ouput/",
    "job_log" : "s3a://bigdata/log/",
    "shutdown_cluster" : false,
    "file_action" : "",
    "submit_job_once_cluster_run" : true,
    "hql" : "",
    "hive_script_path" : ""
  } ],
  "bootstrap_scripts" : [ {
    "name" : "Install zepplin",
    "uri" : "s3a://XXX/zeppelin_install.sh",
    "parameters" : "",
    "nodes" : [ "master" ],
    "active_master" : "false",
    "before_component_start" : "false",
    "start_time" : "1667892101",
    "state" : "IN_PROGRESS",
    "fail_action" : "continue",
    "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
  } ]
}
```

响应示例

状态码： 200

创建集群成功。

```
{
  "cluster_id" : "da1592c2-bb7e-468d-9ac9-83246e95447a",
  "result" : true,
  "msg" : ""
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV11 body = new CreateClusterReqV11();
        List<String> listExecScriptsNodes = new ArrayList<>();
        listExecScriptsNodes.add("master");
        listExecScriptsNodes.add("core");
        listExecScriptsNodes.add("task");
        List<String> listExecScriptsNodes1 = new ArrayList<>();
        listExecScriptsNodes1.add("master");
        listExecScriptsNodes1.add("core");
        listExecScriptsNodes1.add("task");
        List<ScaleScript> listAutoScalingPolicyExecScripts = new ArrayList<>();
        listAutoScalingPolicyExecScripts.add(
            new ScaleScript()
                .withName("before_scale_out")
                .withUri("s3a://XXX/zeppelin_install.sh")
                .withParameters("${mrs_scale_node_num} ${mrs_scale_type} xxx")
                .withNodes(listExecScriptsNodes1)
                .withActiveMaster(true)
                .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
                .withActionStage(ScaleScript.ActionStageEnum.fromValue("before_scale_out"))
        );
        listAutoScalingPolicyExecScripts.add(
            new ScaleScript()
                .withName("after_scale_out")
                .withUri("s3a://XXX/storm_rebalance.sh")
                .withParameters("${mrs_scale_node_hostnames} ${mrs_scale_node_ips}")
                .withNodes(listExecScriptsNodes)
                .withActiveMaster(true)
                .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
```

```
        .withActionStage(ScaleScript.ActionStageEnum.fromValue("after_scale_out"))
);
Trigger triggerRules = new Trigger();
triggerRules.withMetricName("YARNMemoryAvailablePercentage")
    .withMetricValue("70")
    .withComparisonOperator("GT")
    .withEvaluationPeriods(10);
Trigger triggerRules1 = new Trigger();
triggerRules1.withMetricName("YARNMemoryAvailablePercentage")
    .withMetricValue("25")
    .withComparisonOperator("LT")
    .withEvaluationPeriods(10);
List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-expand-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
        .withCoolDownMinutes(5)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules1)
);
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-shrink-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_in"))
        .withCoolDownMinutes(5)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules)
);
List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("9:50")
        .withEndTime("10:20")
        .withMinCapacity(2)
        .withMaxCapacity(3)
);
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("10:20")
        .withEndTime("12:30")
        .withMinCapacity(0)
        .withMaxCapacity(2)
);
AutoScalingPolicy autoScalingPolicyNodeGroups = new AutoScalingPolicy();
autoScalingPolicyNodeGroups.withAutoScalingEnable(true)
    .withMinCapacity(1)
    .withMaxCapacity(3)
    .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
    .withRules(listAutoScalingPolicyRules)
    .withExecScripts(listAutoScalingPolicyExecScripts);
List<NodeGroupV11> listbodyNodeGroups = new ArrayList<>();
listbodyNodeGroups.add(
    new NodeGroupV11()
        .withGroupName("master_node_default_group")
        .withNodeNum(2)
        .withNodeSize("s3.xlarge.2.linux.bigdata")
        .withRootVolumeSize("480")
        .withRootVolumeType("SATA")
        .withDataVolumeType("SATA")
        .withDataVolumeCount(1)
        .withDataVolumeSize(600)
);
listbodyNodeGroups.add(
    new NodeGroupV11()
        .withGroupName("core_node_analysis_group")
        .withNodeNum(3)
```



```
            .withNodeSize("s3.xlarge.2.linux.bigdata")
            .withRootVolumeSize("480")
            .withRootVolumeType("SATA")
            .withDataVolumeType("SATA")
            .withDataVolumeCount(1)
            .withDataVolumeSize(600)
        );
        listbodyNodeGroups.add(
            new NodeGroupV11()
                .withGroupName("task_node_analysis_group")
                .withNodeNum(2)
                .withNodeSize("s3.xlarge.2.linux.bigdata")
                .withRootVolumeSize("480")
                .withRootVolumeType("SATA")
                .withDataVolumeType("SATA")
                .withDataVolumeCount(0)
                .withDataVolumeSize(600)
                .withAutoScalingPolicy(autoScalingPolicyNodeGroups)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages = new ArrayList<>();

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_IN"
));

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_OU
T"));
        List<String> listBootstrapScriptsNodes = new ArrayList<>();
        listBootstrapScriptsNodes.add("master");
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages1 = new ArrayList<>();

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_COMPO
NENT_FIRST_START"));

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_SCALE_I
N"));
        List<String> listBootstrapScriptsNodes1 = new ArrayList<>();
        listBootstrapScriptsNodes1.add("master");
        listBootstrapScriptsNodes1.add("core");
        listBootstrapScriptsNodes1.add("task");
        List<BootstrapScript> listbodyBootstrapScripts = new ArrayList<>();
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Modify os config")
                .withUri("s3a://XXX/modify_os_config.sh")
                .withParameters("param1 param2")
                .withNodes(listBootstrapScriptsNodes1)
                .withActiveMaster(false)
                .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
                .withBeforeComponentStart(true)
                .withStartTime(1667892101L)
                .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
                .withActionStages(listBootstrapScriptsActionStages1)
        );
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Install zepplin")
                .withUri("s3a://XXX/zeppelin_install.sh")
                .withParameters("")
```



```
            .withNodes(listBootstrapScriptsNodes)
            .withActiveMaster(true)
            .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
            .withBeforeComponentStart(false)
            .withStartTime(1667892101L)
            .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
            .withActionStages(listBootstrapScriptsActionStages)
    );
    List<AddJobsReqV11> listbodyAddJobs = new ArrayList<>();
    listbodyAddJobs.add(
        new AddJobsReqV11()
            .withJobType(1)
            .withJobName("tenji111")
            .withJarPath("s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar")
            .withArguments("wordcount")
            .withInput("s3a://bigdata/input/wd_1k/")
            .withOutput("s3a://bigdata/ouput/")
            .withJobLog("s3a://bigdata/log/")
            .withHiveScriptPath("")
            .withHql("")
            .withShutdownCluster(true)
            .withSubmitJobOnceClusterRun(true)
            .withFileAction("")
    );
    List<ComponentAmbV11> listbodyComponentList = new ArrayList<>();
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hadoop")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Spark")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("HBase")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hive")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Presto")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Tez")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hue")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Loader")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Flink")
    );
    body.withNodeGroups(listbodyNodeGroups);
    body.withLoginMode(CreateClusterReqV11.LoginModeEnum.NUMBER_1);
    body.withTags(listbodyTags);
    body.withEnterpriseProjectId("0");
    body.withLogCollection(CreateClusterReqV11.LogCollectionEnum.NUMBER_1);
    body.withClusterType(CreateClusterReqV11.ClusterTypeEnum.NUMBER_0);
    body.withSafeMode(CreateClusterReqV11.SafeModeEnum.NUMBER_0);
    body.withClusterMasterSecret("");
```



```
        body.withClusterAdminSecret("");
        body.withBootstrapScripts(listbodyBootstrapScripts);
        body.withAddJobs(listbodyAddJobs);
        body.withSecurityGroupsId("4820eace-66ad-4f2c-8d46-cf340e3029dd");
        body.withSubnetName("subnet-4b44");
        body.withSubnetId("67984709-e15e-4e86-9886-d76712d4e00a");
        body.withVpcId("4a365717-67be-4f33-80c5-98e98a813af8");
        body.withAvailableZoneId("d573142f24894ef3bd3664de068b44b0");
        body.withComponentList(listbodyComponentList);
        body.withVpc("vpc-4b1c");
        body.withDataCenter("");
        body.withBillingType(CreateClusterReqV11.BillingTypeEnum.NUMBER_12);
        body.withClusterName("mrs_HEbK");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 不使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
```



```
            .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV11 body = new CreateClusterReqV11();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages = new ArrayList<>();

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_IN"
));

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_OU
T"));
        List<String> listBootstrapScriptsNodes = new ArrayList<>();
        listBootstrapScriptsNodes.add("master");
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages1 = new ArrayList<>();

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_COMPO
NENT_FIRST_START"));

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_SCALE_I
N"));
        List<String> listBootstrapScriptsNodes1 = new ArrayList<>();
        listBootstrapScriptsNodes1.add("master");
        listBootstrapScriptsNodes1.add("core");
        listBootstrapScriptsNodes1.add("task");
        List<BootstrapScript> listbodyBootstrapScripts = new ArrayList<>();
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Modifyosconfig")
                .withUri("s3a: //XXX/modify_os_config.sh")
                .withParameters("param1param2")
                .withNodes(listBootstrapScriptsNodes1)
                .withActiveMaster(false)
                .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
                .withBeforeComponentStart(true)
                .withStartTime(1667892101L)
                .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
                .withActionStages(listBootstrapScriptsActionStages1)
        );
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Installzepplin")
                .withUri("s3a: //XXX/zeppelin_install.sh")
                .withParameters("")
                .withNodes(listBootstrapScriptsNodes)
                .withActiveMaster(true)
                .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
                .withBeforeComponentStart(false)
                .withStartTime(1667892101L)
                .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
                .withActionStages(listBootstrapScriptsActionStages)
        );
        List<String> listExecScriptsNodes = new ArrayList<>();
        listExecScriptsNodes.add("master");
        listExecScriptsNodes.add("core");
        listExecScriptsNodes.add("task");
        List<String> listExecScriptsNodes1 = new ArrayList<>();
        listExecScriptsNodes1.add("master");
        listExecScriptsNodes1.add("core");
        listExecScriptsNodes1.add("task");
```



```
List<ScaleScript> listAutoScalingPolicyExecScripts = new ArrayList<>();
listAutoScalingPolicyExecScripts.add(
    new ScaleScript()
        .withName("before_scale_out")
        .withUri("s3a: //XXX/zeppelin_install.sh")
        .withParameters("${mrs_scale_node_num}${mrs_scale_type}xxx")
        .withNodes(listExecScriptsNodes1)
        .withActiveMaster(true)
        .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
        .withActionStage(ScaleScript.ActionStageEnum.fromValue("before_scale_out"))
);
listAutoScalingPolicyExecScripts.add(
    new ScaleScript()
        .withName("after_scale_out")
        .withUri("s3a: //XXX/storm_rebalance.sh")
        .withParameters("${mrs_scale_node_hostnames}${mrs_scale_node_ips}")
        .withNodes(listExecScriptsNodes)
        .withActiveMaster(true)
        .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
        .withActionStage(ScaleScript.ActionStageEnum.fromValue("after_scale_out"))
);
Trigger triggerRules = new Trigger();
triggerRules.withMetricName("YARNMemoryAvailablePercentage")
    .withMetricValue("70")
    .withComparisonOperator("GT")
    .withEvaluationPeriods(10);
Trigger triggerRules1 = new Trigger();
triggerRules1.withMetricName("YARNMemoryAvailablePercentage")
    .withMetricValue("25")
    .withComparisonOperator("LT")
    .withEvaluationPeriods(10);
List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-expand-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
        .withCoolDownMinutes(5)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules1)
);
listAutoScalingPolicyRules.add(
    new Rule()
        .withName("default-shrink-1")
        .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_in"))
        .withCoolDownMinutes(5)
        .withScalingAdjustment(1)
        .withTrigger(triggerRules)
);
List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("9: 50")
        .withEndTime("10: 20")
        .withMinCapacity(2)
        .withMaxCapacity(3)
);
listAutoScalingPolicyResourcesPlans.add(
    new ResourcesPlan()
        .withPeriodType("daily")
        .withStartTime("10: 20")
        .withEndTime("12: 30")
        .withMinCapacity(0)
        .withMaxCapacity(2)
);
AutoScalingPolicy autoScalingPolicyTaskNodeGroups = new AutoScalingPolicy();
autoScalingPolicyTaskNodeGroups.withAutoScalingEnable(true)
    .withMinCapacity(1)
    .withMaxCapacity(3)
```



```
            .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
            .withRules(listAutoScalingPolicyRules)
            .withExecScripts(listAutoScalingPolicyExecScripts);
        List<TaskNodeGroup> listbodyTaskNodeGroups = new ArrayList<>();
        listbodyTaskNodeGroups.add(
            new TaskNodeGroup()
                .withNodeNum(2)
                .withNodeSize("s3.xlarge.2.linux.bigdata")
                .withDataVolumeType(TaskNodeGroup.DataVolumeTypeEnum.fromValue("SATA"))
                .withDataVolumeCount(1)
                .withDataVolumeSize(600)
                .withAutoScalingPolicy(autoScalingPolicyTaskNodeGroups)
        );
        List<AddJobsReqV11> listbodyAddJobs = new ArrayList<>();
        listbodyAddJobs.add(
            new AddJobsReqV11()
                .withJobType(1)
                .withJobName("tenji111")
                .withJarPath("s3a: //bigdata/program/hadoop-mapreduce-examples-2.7.2.jar")
                .withArguments("wordcount")
                .withInput("s3a: //bigdata/input/wd_1k/")
                .withOutput("s3a: //bigdata/ouput/")
                .withJobLog("s3a: //bigdata/log/")
                .withHiveScriptPath("")
                .withHql("")
                .withShutdownCluster(true)
                .withSubmitJobOnceClusterRun(true)
                .withFileAction("")
        );
        List<ComponentAmbV11> listbodyComponentList = new ArrayList<>();
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Hadoop")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Spark")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("HBase")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Hive")
        );
        body.withTags(listbodyTags);
        body.withLogCollection(CreateClusterReqV11.LogCollectionEnum.NUMBER_1);
        body.withClusterType(CreateClusterReqV11.ClusterTypeEnum.NUMBER_0);
        body.withSafeMode(CreateClusterReqV11.SafeModeEnum.NUMBER_0);
        body.withNodePublicCertName("SSHkey-bba1");
        body.withBootstrapScripts(listbodyBootstrapScripts);
        body.withTaskNodeGroups(listbodyTaskNodeGroups);
        body.withCoreDataVolumeCount(2);
        body.withCoreDataVolumeSize(600);

body.withCoreDataVolumeType(CreateClusterReqV11.CoreDataVolumeTypeEnum.fromValue("SATA"));

body.withMasterDataVolumeCount(CreateClusterReqV11.MasterDataVolumeCountEnum.NUMBER_1);
        body.withMasterDataVolumeSize(600);

body.withMasterDataVolumeType(CreateClusterReqV11.MasterDataVolumeTypeEnum.fromValue("SAT
A"));
        body.withAddJobs(listbodyAddJobs);
        body.withSecurityGroupsId("845bece1-fd22-4b45-7a6e-14338c99ee43");
        body.withSubnetName("subnet");
        body.withSubnetId("815bece0-fd22-4b65-8a6e-15788c99ee43");
        body.withVpcId("5b7db34d-3534-4a6e-ac94-023cd36aaf74");
        body.withAvailableZoneId("d573142f24894ef3bd3664de068b44b0");
```



```
        body.withComponentList(listbodyComponentList);
        body.withCoreNodeSize("s1.xlarge.linux.bigdata");
        body.withMasterNodeSize("s3.2xlarge.2.linux.bigdata");
        body.withVpc("vpc1");
        body.withDataCenter("");
        body.withBillingType(CreateClusterReqV11.BillingTypeEnum.NUMBER_12);
        body.withCoreNodeNum(3);
        body.withMasterNodeNum(2);
        body.withClusterName("newcluster");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV11 body = new CreateClusterReqV11();
        List<NodeGroupV11> listbodyNodeGroups = new ArrayList<>();
```



```
        listbodyNodeGroups.add(
            new NodeGroupV11()
                .withGroupName("master_node_default_group")
                .withNodeNum(1)
                .withNodeSize("s3.xlarge.2.linux.bigdata")
                .withRootVolumeSize("480")
                .withRootVolumeType("SATA")
                .withDataVolumeType("SATA")
                .withDataVolumeCount(1)
                .withDataVolumeSize(600)
        );
        listbodyNodeGroups.add(
            new NodeGroupV11()
                .withGroupName("core_node_analysis_group")
                .withNodeNum(1)
                .withNodeSize("s3.xlarge.2.linux.bigdata")
                .withRootVolumeSize("480")
                .withRootVolumeType("SATA")
                .withDataVolumeType("SATA")
                .withDataVolumeCount(1)
                .withDataVolumeSize(600)
        );
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages = new ArrayList<>();

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_IN"
));

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_OU
T"));
        List<String> listBootstrapScriptsNodes = new ArrayList<>();
        listBootstrapScriptsNodes.add("master");
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages1 = new ArrayList<>();

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_COMPO
NENT_FIRST_START"));

listBootstrapScriptsActionStages1.add(BootstrapScript.ActionStagesEnum.fromValue("BEFORE_SCALE_I
N"));
        List<String> listBootstrapScriptsNodes1 = new ArrayList<>();
        listBootstrapScriptsNodes1.add("master");
        listBootstrapScriptsNodes1.add("core");
        listBootstrapScriptsNodes1.add("task");
        List<BootstrapScript> listbodyBootstrapScripts = new ArrayList<>();
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Modify os config")
                .withUri("s3a://XXX/modify_os_config.sh")
                .withParameters("param1 param2")
                .withNodes(listBootstrapScriptsNodes1)
                .withActiveMaster(false)
                .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
                .withBeforeComponentStart(true)
                .withStartTime(1667892101L)
                .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
                .withActionStages(listBootstrapScriptsActionStages1)
        );
        listbodyBootstrapScripts.add(
            new BootstrapScript()
```



```
            .withName("Install zepplin")
            .withUri("s3a://XXX/zeppelin_install.sh")
            .withParameters("")
            .withNodes(listBootstrapScriptsNodes)
            .withActiveMaster(true)
            .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
            .withBeforeComponentStart(false)
            .withStartTime(1667892101L)
            .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
            .withActionStages(listBootstrapScriptsActionStages)
    );
    List<AddJobsReqV11> listbodyAddJobs = new ArrayList<>();
    listbodyAddJobs.add(
        new AddJobsReqV11()
            .withJobType(1)
            .withJobName("tenji111")
            .withJarPath("s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar")
            .withArguments("wordcount")
            .withInput("s3a://bigdata/input/wd_1k/")
            .withOutput("s3a://bigdata/ouput/")
            .withJobLog("s3a://bigdata/log/")
            .withHiveScriptPath("")
            .withHql("")
            .withShutdownCluster(true)
            .withSubmitJobOnceClusterRun(true)
            .withFileAction("")
    );
    List<ComponentAmbV11> listbodyComponentList = new ArrayList<>();
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hadoop")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Spark")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("HBase")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hive")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Presto")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Tez")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Hue")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Loader")
    );
    listbodyComponentList.add(
        new ComponentAmbV11()
            .withComponentName("Flink")
    );
    body.withNodeGroups(listbodyNodeGroups);
    body.withLoginMode(CreateClusterReqV11.LoginModeEnum.NUMBER_1);
    body.withTags(listbodyTags);
    body.withEnterpriseProjectId("0");
    body.withLogCollection(CreateClusterReqV11.LogCollectionEnum.NUMBER_1);
```



```
        body.withClusterType(CreateClusterReqV11.ClusterTypeEnum.NUMBER_0);
        body.withSafeMode(CreateClusterReqV11.SafeModeEnum.NUMBER_0);
        body.withClusterMasterSecret("");
        body.withClusterAdminSecret("");
        body.withBootstrapScripts(listbodyBootstrapScripts);
        body.withAddJobs(listbodyAddJobs);
        body.withSecurityGroupsId("4820eace-66ad-4f2c-8d46-cf340e3029dd");
        body.withSubnetName("subnet-4b44");
        body.withSubnetId("67984709-e15e-4e86-9886-d76712d4e00a");
        body.withVpcId("4a365717-67be-4f33-80c5-98e98a813af8");
        body.withAvailableZoneId("d573142f24894ef3bd3664de068b44b0");
        body.withComponentList(listbodyComponentList);
        body.withVpc("vpc-4b1c");
        body.withDataCenter("");
        body.withBillingType(CreateClusterReqV11.BillingTypeEnum.NUMBER_12);
        body.withClusterName("mrs_HEbK");
        body.withClusterVersion("MRS 3.1.0");
        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());
        }
    }
}
```

- 不使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

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");
        String projectId = "{project_id}";

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

```
        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateClusterRequest request = new CreateClusterRequest();
        CreateClusterReqV11 body = new CreateClusterReqV11();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        List<BootstrapScript.ActionStagesEnum> listBootstrapScriptsActionStages = new ArrayList<>();

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_IN"
));

listBootstrapScriptsActionStages.add(BootstrapScript.ActionStagesEnum.fromValue("AFTER_SCALE_OU
T"));
        List<String> listBootstrapScriptsNodes = new ArrayList<>();
        listBootstrapScriptsNodes.add("master");
        List<BootstrapScript> listbodyBootstrapScripts = new ArrayList<>();
        listbodyBootstrapScripts.add(
            new BootstrapScript()
                .withName("Install zepplin")
                .withUri("s3a://XXX/zeppelin_install.sh")
                .withParameters("")
                .withNodes(listBootstrapScriptsNodes)
                .withActiveMaster(false)
                .withFailAction(BootstrapScript.FailActionEnum.fromValue("continue"))
                .withBeforeComponentStart(false)
                .withStartTime(1667892101L)
                .withState(BootstrapScript.StateEnum.fromValue("IN_PROGRESS"))
                .withActionStages(listBootstrapScriptsActionStages)
        );
        List<AddJobsReqV11> listbodyAddJobs = new ArrayList<>();
        listbodyAddJobs.add(
            new AddJobsReqV11()
                .withJobType(1)
                .withJobName("tenji111")
                .withJarPath("s3a://bigdata/program/hadoop-mapreduce-examples-XXX.jar")
                .withArguments("wordcount")
                .withInput("s3a://bigdata/input/wd_1k/")
                .withOutput("s3a://bigdata/ouput/")
                .withJobLog("s3a://bigdata/log/")
                .withHiveScriptPath("")
                .withHql("")
                .withShutdownCluster(false)
                .withSubmitJobOnceClusterRun(true)
                .withFileAction("")
        );
        List<ComponentAmbV11> listbodyComponentList = new ArrayList<>();
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Hadoop")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Spark")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("HBase")
        );
```



```
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Hive")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Presto")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Tez")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Hue")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Loader")
        );
        listbodyComponentList.add(
            new ComponentAmbV11()
                .withComponentName("Flink")
        );
        body.withLoginMode(CreateClusterReqV11.LoginModeEnum.NUMBER_1);
        body.withTags(listbodyTags);
        body.withEnterpriseProjectId("0");
        body.withLogCollection(CreateClusterReqV11.LogCollectionEnum.NUMBER_1);
        body.withClusterType(CreateClusterReqV11.ClusterTypeEnum.NUMBER_0);
        body.withSafeMode(CreateClusterReqV11.SafeModeEnum.NUMBER_0);
        body.withClusterAdminSecret("******");
        body.withNodePublicCertName("SSHkey-bba1");
        body.withBootstrapScripts(listbodyBootstrapScripts);
        body.withCoreDataVolumeCount(1);
        body.withCoreDataVolumeSize(600);

body.withCoreDataVolumeType(CreateClusterReqV11.CoreDataVolumeTypeEnum.fromValue("SATA"));

body.withMasterDataVolumeCount(CreateClusterReqV11.MasterDataVolumeCountEnum.NUMBER_1);
        body.withMasterDataVolumeSize(600);

body.withMasterDataVolumeType(CreateClusterReqV11.MasterDataVolumeTypeEnum.fromValue("SAT
A"));
        body.withAddJobs(listbodyAddJobs);
        body.withSecurityGroupsId("");
        body.withSubnetName("subnet");
        body.withSubnetId("815bece0-fd22-4b65-8a6e-15788c99ee43");
        body.withVpcId("5b7db34d-3534-4a6e-ac94-023cd36aaf74");
        body.withAvailableZoneId("d573142f24894ef3bd3664de068b44b0");
        body.withComponentList(listbodyComponentList);
        body.withCoreNodeSize("s1.xlarge.linux.bigdata");
        body.withMasterNodeSize("s3.2xlarge.2.linux.bigdata");
        body.withVpc("vpc1");
        body.withDataCenter("");
        body.withBillingType(CreateClusterReqV11.BillingTypeEnum.NUMBER_12);
        body.withCoreNodeNum(1);
        body.withMasterNodeNum(1);
        body.withClusterName("newcluster");
        body.withClusterVersion("MRS 3.1.0");
        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

- 使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listNodesExecScripts = [
            "master",
            "core",
            "task"
        ]
        listNodesExecScripts1 = [
            "master",
            "core",
            "task"
        ]
        listExecScriptsAutoScalingPolicy = [
            ScaleScript(
                name="before_scale_out",
                uri="s3a://XXX/zeppelin_install.sh",
                parameters="${mrs_scale_node_num} ${mrs_scale_type} xxx",
                nodes=listNodesExecScripts1,
                active_master=True,
                fail_action="continue",
                action_stage="before_scale_out"
            ),
            ScaleScript(
                name="after_scale_out",
                uri="s3a://XXX/storm_rebalance.sh",
                parameters="${mrs_scale_node_hostnames} ${mrs_scale_node_ips}",
                nodes=listNodesExecScripts,
                active_master=True,
                fail_action="continue",
                action_stage="after_scale_out"
```



```
        )
    ]
    triggerRules = Trigger(
        metric_name="YARNMemoryAvailablePercentage",
        metric_value="70",
        comparison_operator="GT",
        evaluation_periods=10
    )
    triggerRules1 = Trigger(
        metric_name="YARNMemoryAvailablePercentage",
        metric_value="25",
        comparison_operator="LT",
        evaluation_periods=10
    )
    listRulesAutoScalingPolicy = [
        Rule(
            name="default-expand-1",
            adjustment_type="scale_out",
            cool_down_minutes=5,
            scaling_adjustment=1,
            trigger=triggerRules1
        ),
        Rule(
            name="default-shrink-1",
            adjustment_type="scale_in",
            cool_down_minutes=5,
            scaling_adjustment=1,
            trigger=triggerRules
        )
    ]
    listResourcesPlansAutoScalingPolicy = [
        ResourcesPlan(
            period_type="daily",
            start_time="9:50",
            end_time="10:20",
            min_capacity=2,
            max_capacity=3
        ),
        ResourcesPlan(
            period_type="daily",
            start_time="10:20",
            end_time="12:30",
            min_capacity=0,
            max_capacity=2
        )
    ]
    autoScalingPolicyNodeGroups = AutoScalingPolicy(
        auto_scaling_enable=True,
        min_capacity=1,
        max_capacity=3,
        resources_plans=listResourcesPlansAutoScalingPolicy,
        rules=listRulesAutoScalingPolicy,
        exec_scripts=listExecScriptsAutoScalingPolicy
    )
    listNodeGroupsbody = [
        NodeGroupV11(
            group_name="master_node_default_group",
            node_num=2,
            node_size="s3.xlarge.2.linux.bigdata",
            root_volume_size="480",
            root_volume_type="SATA",
            data_volume_type="SATA",
            data_volume_count=1,
            data_volume_size=600
        ),
        NodeGroupV11(
            group_name="core_node_analysis_group",
            node_num=3,
            node_size="s3.xlarge.2.linux.bigdata",
```



```
        root_volume_size="480",
        root_volume_type="SATA",
        data_volume_type="SATA",
        data_volume_count=1,
        data_volume_size=600
    ),
    NodeGroupV11(
        group_name="task_node_analysis_group",
        node_num=2,
        node_size="s3.xlarge.2.linux.bigdata",
        root_volume_size="480",
        root_volume_type="SATA",
        data_volume_type="SATA",
        data_volume_count=0,
        data_volume_size=600,
        auto_scaling_policy=autoScalingPolicyNodeGroups
    )
]
listTagsbody = [
    Tag(
        key="key1",
        value="value1"
    ),
    Tag(
        key="key2",
        value="value2"
    )
]
listActionStagesBootstrapScripts = [
    "AFTER_SCALE_IN",
    "AFTER_SCALE_OUT"
]
listNodesBootstrapScripts = [
    "master"
]
listActionStagesBootstrapScripts1 = [
    "BEFORE_COMPONENT_FIRST_START",
    "BEFORE_SCALE_IN"
]
listNodesBootstrapScripts1 = [
    "master",
    "core",
    "task"
]
listBootstrapScriptsbody = [
    BootstrapScript(
        name="Modify os config",
        uri="s3a://XXX/modify_os_config.sh",
        parameters="param1 param2",
        nodes=listNodesBootstrapScripts1,
        active_master=False,
        fail_action="continue",
        before_component_start=True,
        start_time=1667892101,
        state="IN_PROGRESS",
        action_stages=listActionStagesBootstrapScripts1
    ),
    BootstrapScript(
        name="Install zepplin",
        uri="s3a://XXX/zeppelin_install.sh",
        parameters="",
        nodes=listNodesBootstrapScripts,
        active_master=True,
        fail_action="continue",
        before_component_start=False,
        start_time=1667892101,
        state="IN_PROGRESS",
        action_stages=listActionStagesBootstrapScripts
    )
```



```
    ]
    listAddJobsbody = [
        AddJobsReqV11(
            job_type=1,
            job_name="tenji111",
            jar_path="s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
            arguments="wordcount",
            input="s3a://bigdata/input/wd_1k/",
            output="s3a://bigdata/ouput/",
            job_log="s3a://bigdata/log/",
            hive_script_path="",
            hql="",
            shutdown_cluster=True,
            submit_job_once_cluster_run=True,
            file_action=""
        )
    ]
    listComponentListbody = [
        ComponentAmbV11(
            component_name="Hadoop"
        ),
        ComponentAmbV11(
            component_name="Spark"
        ),
        ComponentAmbV11(
            component_name="HBase"
        ),
        ComponentAmbV11(
            component_name="Hive"
        ),
        ComponentAmbV11(
            component_name="Presto"
        ),
        ComponentAmbV11(
            component_name="Tez"
        ),
        ComponentAmbV11(
            component_name="Hue"
        ),
        ComponentAmbV11(
            component_name="Loader"
        ),
        ComponentAmbV11(
            component_name="Flink"
        )
    ]
    request.body = CreateClusterReqV11(
        node_groups=listNodeGroupsbody,
        login_mode=1,
        tags=listTagsbody,
        enterprise_project_id="0",
        log_collection=1,
        cluster_type=0,
        safe_mode=0,
        cluster_master_secret="",
        cluster_admin_secret="",
        bootstrap_scripts=listBootstrapScriptsbody,
        add_jobs=listAddJobsbody,
        security_groups_id="4820eace-66ad-4f2c-8d46-cf340e3029dd",
        subnet_name="subnet-4b44",
        subnet_id="67984709-e15e-4e86-9886-d76712d4e00a",
        vpc_id="4a365717-67be-4f33-80c5-98e98a813af8",
        available_zone_id="d573142f24894ef3bd3664de068b44b0",
        component_list=listComponentListbody,
        vpc="vpc-4b1c",
        data_center="",
        billing_type=12,
        cluster_name="mrs_HEbK",
        cluster_version="MRS 3.1.0"
```

```
        )
        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)
```

- 不使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        listActionStagesBootstrapScripts = [
            "AFTER_SCALE_IN",
            "AFTER_SCALE_OUT"
        ]
        listNodesBootstrapScripts = [
            "master"
        ]
        listActionStagesBootstrapScripts1 = [
            "BEFORE_COMPONENT_FIRST_START",
            "BEFORE_SCALE_IN"
        ]
        listNodesBootstrapScripts1 = [
            "master",
            "core",
            "task"
        ]
        listBootstrapScriptsbody = [
            BootstrapScript(
                name="Modifyosconfig",
                uri="s3a: //XXX/modify_os_config.sh",
```



```
        parameters="param1param2",
        nodes=listNodesBootstrapScripts1,
        active_master=False,
        fail_action="continue",
        before_component_start=True,
        start_time=1667892101,
        state="IN_PROGRESS",
        action_stages=listActionStagesBootstrapScripts1
    ),
    BootstrapScript(
        name="Installzepplin",
        uri="s3a: //XXX/zeppelin_install.sh",
        parameters="",
        nodes=listNodesBootstrapScripts,
        active_master=True,
        fail_action="continue",
        before_component_start=False,
        start_time=1667892101,
        state="IN_PROGRESS",
        action_stages=listActionStagesBootstrapScripts
    )
]
listNodesExecScripts = [
    "master",
    "core",
    "task"
]
listNodesExecScripts1 = [
    "master",
    "core",
    "task"
]
listExecScriptsAutoScalingPolicy = [
    ScaleScript(
        name="before_scale_out",
        uri="s3a: //XXX/zeppelin_install.sh",
        parameters="${mrs_scale_node_num}${mrs_scale_type}xxx",
        nodes=listNodesExecScripts1,
        active_master=True,
        fail_action="continue",
        action_stage="before_scale_out"
    ),
    ScaleScript(
        name="after_scale_out",
        uri="s3a: //XXX/storm_rebalance.sh",
        parameters="${mrs_scale_node_hostnames}${mrs_scale_node_ips}",
        nodes=listNodesExecScripts,
        active_master=True,
        fail_action="continue",
        action_stage="after_scale_out"
    )
]
triggerRules = Trigger(
    metric_name="YARNMemoryAvailablePercentage",
    metric_value="70",
    comparison_operator="GT",
    evaluation_periods=10
)
triggerRules1 = Trigger(
    metric_name="YARNMemoryAvailablePercentage",
    metric_value="25",
    comparison_operator="LT",
    evaluation_periods=10
)
listRulesAutoScalingPolicy = [
    Rule(
        name="default-expand-1",
        adjustment_type="scale_out",
        cool_down_minutes=5,
```



```
        scaling_adjustment=1,
        trigger=triggerRules1
    ),
    Rule(
        name="default-shrink-1",
        adjustment_type="scale_in",
        cool_down_minutes=5,
        scaling_adjustment=1,
        trigger=triggerRules
    )
]
listResourcesPlansAutoScalingPolicy = [
    ResourcesPlan(
        period_type="daily",
        start_time="9: 50",
        end_time="10: 20",
        min_capacity=2,
        max_capacity=3
    ),
    ResourcesPlan(
        period_type="daily",
        start_time="10: 20",
        end_time="12: 30",
        min_capacity=0,
        max_capacity=2
    )
]
autoScalingPolicyTaskNodeGroups = AutoScalingPolicy(
    auto_scaling_enable=True,
    min_capacity=1,
    max_capacity=3,
    resources_plans=listResourcesPlansAutoScalingPolicy,
    rules=listRulesAutoScalingPolicy,
    exec_scripts=listExecScriptsAutoScalingPolicy
)
listTaskNodeGroupsbody = [
    TaskNodeGroup(
        node_num=2,
        node_size="s3.xlarge.2.linux.bigdata",
        data_volume_type="SATA",
        data_volume_count=1,
        data_volume_size=600,
        auto_scaling_policy=autoScalingPolicyTaskNodeGroups
    )
]
listAddJobsbody = [
    AddJobsReqV11(
        job_type=1,
        job_name="tenji111",
        jar_path="s3a: //bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
        arguments="wordcount",
        input="s3a: //bigdata/input/wd_1k/",
        output="s3a: //bigdata/ouput/",
        job_log="s3a: //bigdata/log/",
        hive_script_path="",
        hql="",
        shutdown_cluster=True,
        submit_job_once_cluster_run=True,
        file_action=""
    )
]
listComponentListbody = [
    ComponentAmbV11(
        component_name="Hadoop"
    ),
    ComponentAmbV11(
        component_name="Spark"
    ),
    ComponentAmbV11(
```



```
            component_name="HBase"
        ),
        ComponentAmbV11(
            component_name="Hive"
        )
    ]
    request.body = CreateClusterReqV11(
        tags=listTagsbody,
        log_collection=1,
        cluster_type=0,
        safe_mode=0,
        node_public_cert_name="SSHkey-bba1",
        bootstrap_scripts=listBootstrapScriptsbody,
        task_node_groups=listTaskNodeGroupsbody,
        core_data_volume_count=2,
        core_data_volume_size=600,
        core_data_volume_type="SATA",
        master_data_volume_count=1,
        master_data_volume_size=600,
        master_data_volume_type="SATA",
        add_jobs=listAddJobsbody,
        security_groups_id="845bece1-fd22-4b45-7a6e-14338c99ee43",
        subnet_name="subnet",
        subnet_id="815bece0-fd22-4b65-8a6e-15788c99ee43",
        vpc_id="5b7db34d-3534-4a6e-ac94-023cd36aaf74",
        available_zone_id="d573142f24894ef3bd3664de068b44b0",
        component_list=listComponentListbody,
        core_node_size="s1.xlarge.linux.bigdata",
        master_node_size="s3.2xlarge.2.linux.bigdata",
        vpc="vpc1",
        data_center="",
        billing_type=12,
        core_node_num=3,
        master_node_num=2,
        cluster_name="newcluster",
        cluster_version="MRS 3.1.0"
    )
    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)
```

- 使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

    client = MrsClient.new_builder() \
```

```
    .with_credentials(credentials) \
    .with_region(MrsRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateClusterRequest()
    listNodeGroupsbody = [
        NodeGroupV11(
            group_name="master_node_default_group",
            node_num=1,
            node_size="s3.xlarge.2.linux.bigdata",
            root_volume_size="480",
            root_volume_type="SATA",
            data_volume_type="SATA",
            data_volume_count=1,
            data_volume_size=600
        ),
        NodeGroupV11(
            group_name="core_node_analysis_group",
            node_num=1,
            node_size="s3.xlarge.2.linux.bigdata",
            root_volume_size="480",
            root_volume_type="SATA",
            data_volume_type="SATA",
            data_volume_count=1,
            data_volume_size=600
        )
    ]
    listTagsbody = [
        Tag(
            key="key1",
            value="value1"
        ),
        Tag(
            key="key2",
            value="value2"
        )
    ]
    listActionStagesBootstrapScripts = [
        "AFTER_SCALE_IN",
        "AFTER_SCALE_OUT"
    ]
    listNodesBootstrapScripts = [
        "master"
    ]
    listActionStagesBootstrapScripts1 = [
        "BEFORE_COMPONENT_FIRST_START",
        "BEFORE_SCALE_IN"
    ]
    listNodesBootstrapScripts1 = [
        "master",
        "core",
        "task"
    ]
    listBootstrapScriptsbody = [
        BootstrapScript(
            name="Modify os config",
            uri="s3a://XXX/modify_os_config.sh",
            parameters="param1 param2",
            nodes=listNodesBootstrapScripts1,
            active_master=False,
            fail_action="continue",
            before_component_start=True,
            start_time=1667892101,
            state="IN_PROGRESS",
            action_stages=listActionStagesBootstrapScripts1
        ),
        BootstrapScript(
            name="Install zepplin",
```



```
            uri="s3a://XXX/zeppelin_install.sh",
            parameters="",
            nodes=listNodesBootstrapScripts,
            active_master=True,
            fail_action="continue",
            before_component_start=False,
            start_time=1667892101,
            state="IN_PROGRESS",
            action_stages=listActionStagesBootstrapScripts
        )
    ]
    listAddJobsbody = [
        AddJobsReqV11(
            job_type=1,
            job_name="tenji111",
            jar_path="s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar",
            arguments="wordcount",
            input="s3a://bigdata/input/wd_1k/",
            output="s3a://bigdata/ouput/",
            job_log="s3a://bigdata/log/",
            hive_script_path="",
            hql="",
            shutdown_cluster=True,
            submit_job_once_cluster_run=True,
            file_action=""
        )
    ]
    listComponentListbody = [
        ComponentAmbV11(
            component_name="Hadoop"
        ),
        ComponentAmbV11(
            component_name="Spark"
        ),
        ComponentAmbV11(
            component_name="HBase"
        ),
        ComponentAmbV11(
            component_name="Hive"
        ),
        ComponentAmbV11(
            component_name="Presto"
        ),
        ComponentAmbV11(
            component_name="Tez"
        ),
        ComponentAmbV11(
            component_name="Hue"
        ),
        ComponentAmbV11(
            component_name="Loader"
        ),
        ComponentAmbV11(
            component_name="Flink"
        )
    ]
    request.body = CreateClusterReqV11(
        node_groups=listNodeGroupsbody,
        login_mode=1,
        tags=listTagsbody,
        enterprise_project_id="0",
        log_collection=1,
        cluster_type=0,
        safe_mode=0,
        cluster_master_secret="",
        cluster_admin_secret="",
        bootstrap_scripts=listBootstrapScriptsbody,
        add_jobs=listAddJobsbody,
        security_groups_id="4820eace-66ad-4f2c-8d46-cf340e3029dd",
```



```
            subnet_name="subnet-4b44",
            subnet_id="67984709-e15e-4e86-9886-d76712d4e00a",
            vpc_id="4a365717-67be-4f33-80c5-98e98a813af8",
            available_zone_id="d573142f24894ef3bd3664de068b44b0",
            component_list=listComponentListbody,
            vpc="vpc-4b1c",
            data_center="",
            billing_type=12,
            cluster_name="mrs_HEbK",
            cluster_version="MRS 3.1.0"
        )
        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)
```

- 不使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateClusterRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        listActionStagesBootstrapScripts = [
            "AFTER_SCALE_IN",
            "AFTER_SCALE_OUT"
        ]
        listNodesBootstrapScripts = [
            "master"
        ]
        listBootstrapScriptsbody = [
            BootstrapScript(
                name="Install zepplin",
```

```
            uri="s3a://XXX/zeppelin_install.sh",
            parameters="",
            nodes=listNodesBootstrapScripts,
            active_master=False,
            fail_action="continue",
            before_component_start=False,
            start_time=1667892101,
            state="IN_PROGRESS",
            action_stages=listActionStagesBootstrapScripts
        )
    ]
    listAddJobsbody = [
        AddJobsReqV11(
            job_type=1,
            job_name="tenji111",
            jar_path="s3a://bigdata/program/hadoop-mapreduce-examples-XXX.jar",
            arguments="wordcount",
            input="s3a://bigdata/input/wd_1k/",
            output="s3a://bigdata/ouput/",
            job_log="s3a://bigdata/log/",
            hive_script_path="",
            hql="",
            shutdown_cluster=False,
            submit_job_once_cluster_run=True,
            file_action=""
        )
    ]
    listComponentListbody = [
        ComponentAmbV11(
            component_name="Hadoop"
        ),
        ComponentAmbV11(
            component_name="Spark"
        ),
        ComponentAmbV11(
            component_name="HBase"
        ),
        ComponentAmbV11(
            component_name="Hive"
        ),
        ComponentAmbV11(
            component_name="Presto"
        ),
        ComponentAmbV11(
            component_name="Tez"
        ),
        ComponentAmbV11(
            component_name="Hue"
        ),
        ComponentAmbV11(
            component_name="Loader"
        ),
        ComponentAmbV11(
            component_name="Flink"
        )
    ]
    request.body = CreateClusterReqV11(
        login_mode=1,
        tags=listTagsbody,
        enterprise_project_id="0",
        log_collection=1,
        cluster_type=0,
        safe_mode=0,
        cluster_admin_secret="******",
        node_public_cert_name="SSHkey-bba1",
        bootstrap_scripts=listBootstrapScriptsbody,
        core_data_volume_count=1,
        core_data_volume_size=600,
        core_data_volume_type="SATA",
```



```
            master_data_volume_count=1,
            master_data_volume_size=600,
            master_data_volume_type="SATA",
            add_jobs=listAddJobsbody,
            security_groups_id="",
            subnet_name="subnet",
            subnet_id="815bece0-fd22-4b65-8a6e-15788c99ee43",
            vpc_id="5b7db34d-3534-4a6e-ac94-023cd36aaf74",
            available_zone_id="d573142f24894ef3bd3664de068b44b0",
            component_list=listComponentListbody,
            core_node_size="s1.xlarge.linux.bigdata",
            master_node_size="s3.2xlarge.2.linux.bigdata",
            vpc="vpc1",
            data_center="",
            billing_type=12,
            core_node_num=1,
            master_node_num=1,
            cluster_name="newcluster",
            cluster_version="MRS 3.1.0"
        )
        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

- 使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listNodesExecScripts = []string{
        "master",
```



```
    "core",
    "task",
}
var listNodesExecScripts1 = []string{
    "master",
    "core",
    "task",
}
parametersExecScripts:= "${mrs_scale_node_num} ${mrs_scale_type} xxx"
activeMasterExecScripts:= true
parametersExecScripts1:= "${mrs_scale_node_hostnames} ${mrs_scale_node_ips}"
activeMasterExecScripts1:= true
var listExecScriptsAutoScalingPolicy = []model.ScaleScript{
    {
        Name: "before_scale_out",
        Uri: "s3a://XXX/zeppelin_install.sh",
        Parameters: &parametersExecScripts,
        Nodes: listNodesExecScripts1,
        ActiveMaster: &activeMasterExecScripts,
        FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
        ActionStage: model.GetScaleScriptActionStageEnum().BEFORE_SCALE_OUT,
    },
    {
        Name: "after_scale_out",
        Uri: "s3a://XXX/storm_rebalance.sh",
        Parameters: &parametersExecScripts1,
        Nodes: listNodesExecScripts,
        ActiveMaster: &activeMasterExecScripts1,
        FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
        ActionStage: model.GetScaleScriptActionStageEnum().AFTER_SCALE_OUT,
    },
}
comparisonOperatorTrigger:= "GT"
triggerRules := &model.Trigger{
    MetricName: "YARNMemoryAvailablePercentage",
    MetricValue: "70",
    ComparisonOperator: &comparisonOperatorTrigger,
    EvaluationPeriods: int32(10),
}
comparisonOperatorTrigger1:= "LT"
triggerRules1 := &model.Trigger{
    MetricName: "YARNMemoryAvailablePercentage",
    MetricValue: "25",
    ComparisonOperator: &comparisonOperatorTrigger1,
    EvaluationPeriods: int32(10),
}
var listRulesAutoScalingPolicy = []model.Rule{
    {
        Name: "default-expand-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
        CoolDownMinutes: int32(5),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules1,
    },
    {
        Name: "default-shrink-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_IN,
        CoolDownMinutes: int32(5),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules,
    },
}
var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
    {
        PeriodType: "daily",
        StartTime: "9:50",
        EndTime: "10:20",
        MinCapacity: int32(2),
        MaxCapacity: int32(3),
```



```
    },
    {
        PeriodType: "daily",
        StartTime: "10:20",
        EndTime: "12:30",
        MinCapacity: int32(0),
        MaxCapacity: int32(2),
    },
}
autoScalingPolicyNodeGroups := &model.AutoScalingPolicy{
    AutoScalingEnable: true,
    MinCapacity: int32(1),
    MaxCapacity: int32(3),
    ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
    Rules: &listRulesAutoScalingPolicy,
    ExecScripts: &listExecScriptsAutoScalingPolicy,
}
rootVolumeSizeNodeGroups:= "480"
rootVolumeTypeNodeGroups:= "SATA"
dataVolumeTypeNodeGroups:= "SATA"
dataVolumeCountNodeGroups:= int32(1)
dataVolumeSizeNodeGroups:= int32(600)
rootVolumeSizeNodeGroups1:= "480"
rootVolumeTypeNodeGroups1:= "SATA"
dataVolumeTypeNodeGroups1:= "SATA"
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeSizeNodeGroups1:= int32(600)
rootVolumeSizeNodeGroups2:= "480"
rootVolumeTypeNodeGroups2:= "SATA"
dataVolumeTypeNodeGroups2:= "SATA"
dataVolumeCountNodeGroups2:= int32(0)
dataVolumeSizeNodeGroups2:= int32(600)
var listNodeGroupsbody = []model.NodeGroupV11{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(2),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        RootVolumeSize: &rootVolumeSizeNodeGroups,
        RootVolumeType: &rootVolumeTypeNodeGroups,
        DataVolumeType: &dataVolumeTypeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups,
        DataVolumeSize: &dataVolumeSizeNodeGroups,
    },
    {
        GroupName: "core_node_analysis_group",
        NodeNum: int32(3),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        RootVolumeSize: &rootVolumeSizeNodeGroups1,
        RootVolumeType: &rootVolumeTypeNodeGroups1,
        DataVolumeType: &dataVolumeTypeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
        DataVolumeSize: &dataVolumeSizeNodeGroups1,
    },
    {
        GroupName: "task_node_analysis_group",
        NodeNum: int32(2),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        RootVolumeSize: &rootVolumeSizeNodeGroups2,
        RootVolumeType: &rootVolumeTypeNodeGroups2,
        DataVolumeType: &dataVolumeTypeNodeGroups2,
        DataVolumeCount: &dataVolumeCountNodeGroups2,
        DataVolumeSize: &dataVolumeSizeNodeGroups2,
        AutoScalingPolicy: autoScalingPolicyNodeGroups,
    },
}
var listTagsbody = []model.Tag{
    {
        Key: "key1",
        Value: "value1",
```



```
    },
    {
        Key: "key2",
        Value: "value2",
    },
}
var listActionStagesBootstrapScripts = []model.BootstrapScriptActionStages{
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_IN,
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_OUT,
}
var listNodesBootstrapScripts = []string{
    "master",
}
var listActionStagesBootstrapScripts1 = []model.BootstrapScriptActionStages{
    model.GetBootstrapScriptActionStagesEnum().BEFORE_COMPONENT_FIRST_START,
    model.GetBootstrapScriptActionStagesEnum().BEFORE_SCALE_IN,
}
var listNodesBootstrapScripts1 = []string{
    "master",
    "core",
    "task",
}
parametersBootstrapScripts:= "param1 param2"
activeMasterBootstrapScripts:= false
beforeComponentStartBootstrapScripts:= true
startTimeBootstrapScripts:= int64(1667892101)
stateBootstrapScripts:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
parametersBootstrapScripts1:= ""
activeMasterBootstrapScripts1:= true
beforeComponentStartBootstrapScripts1:= false
startTimeBootstrapScripts1:= int64(1667892101)
stateBootstrapScripts1:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
var listBootstrapScriptsbody = []model.BootstrapScript{
    {
        Name: "Modify os config",
        Uri: "s3a://XXX/modify_os_config.sh",
        Parameters: &parametersBootstrapScripts,
        Nodes: listNodesBootstrapScripts1,
        ActiveMaster: &activeMasterBootstrapScripts,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts,
        StartTime: &startTimeBootstrapScripts,
        State: &stateBootstrapScripts,
        ActionStages: &listActionStagesBootstrapScripts1,
    },
    {
        Name: "Install zepplin",
        Uri: "s3a://XXX/zeppelin_install.sh",
        Parameters: &parametersBootstrapScripts1,
        Nodes: listNodesBootstrapScripts,
        ActiveMaster: &activeMasterBootstrapScripts1,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts1,
        StartTime: &startTimeBootstrapScripts1,
        State: &stateBootstrapScripts1,
        ActionStages: &listActionStagesBootstrapScripts,
    },
}
jarPathAddJobs:= "s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar"
argumentsAddJobs:= "wordcount"
inputAddJobs:= "s3a://bigdata/input/wd_1k/"
outputAddJobs:= "s3a://bigdata/ouput/"
jobLogAddJobs:= "s3a://bigdata/log/"
hiveScriptPathAddJobs:= ""
hqlAddJobs:= ""
shutdownClusterAddJobs:= true
fileActionAddJobs:= ""
var listAddJobsbody = []model.AddJobsReqV11{
    {
```



```
        JobType: int32(1),
        JobName: "tenji111",
        JarPath: &jarPathAddJobs,
        Arguments: &argumentsAddJobs,
        Input: &inputAddJobs,
        Output: &outputAddJobs,
        JobLog: &jobLogAddJobs,
        HiveScriptPath: &hiveScriptPathAddJobs,
        Hql: &hqlAddJobs,
        ShutdownCluster: &shutdownClusterAddJobs,
        SubmitJobOnceClusterRun: true,
        FileAction: &fileActionAddJobs,
    },
}
var listComponentListbody = []model.ComponentAmbV11{
    {
        ComponentName: "Hadoop",
    },
    {
        ComponentName: "Spark",
    },
    {
        ComponentName: "HBase",
    },
    {
        ComponentName: "Hive",
    },
    {
        ComponentName: "Presto",
    },
    {
        ComponentName: "Tez",
    },
    {
        ComponentName: "Hue",
    },
    {
        ComponentName: "Loader",
    },
    {
        ComponentName: "Flink",
    },
}
loginModeCreateClusterReqV11:= model.GetCreateClusterReqV11LoginModeEnum().E_1
enterpriseProjectIdCreateClusterReqV11:= "0"
logCollectionCreateClusterReqV11:= model.GetCreateClusterReqV11LogCollectionEnum().E_1
clusterTypeCreateClusterReqV11:= model.GetCreateClusterReqV11ClusterTypeEnum().E_0
clusterMasterSecretCreateClusterReqV11:= ""
clusterAdminSecretCreateClusterReqV11:= ""
securityGroupsIdCreateClusterReqV11:= "4820eace-66ad-4f2c-8d46-cf340e3029dd"
request.Body = &model.CreateClusterReqV11{
    NodeGroups: &listNodeGroupsbody,
    LoginMode: &loginModeCreateClusterReqV11,
    Tags: &listTagsbody,
    EnterpriseProjectId: &enterpriseProjectIdCreateClusterReqV11,
    LogCollection: &logCollectionCreateClusterReqV11,
    ClusterType: &clusterTypeCreateClusterReqV11,
    SafeMode: model.GetCreateClusterReqV11SafeModeEnum().E_0,
    ClusterMasterSecret: &clusterMasterSecretCreateClusterReqV11,
    ClusterAdminSecret: &clusterAdminSecretCreateClusterReqV11,
    BootstrapScripts: &listBootstrapScriptsbody,
    AddJobs: &listAddJobsbody,
    SecurityGroupsId: &securityGroupsIdCreateClusterReqV11,
    SubnetName: "subnet-4b44",
    SubnetId: "67984709-e15e-4e86-9886-d76712d4e00a",
    VpcId: "4a365717-67be-4f33-80c5-98e98a813af8",
    AvailableZoneId: "d573142f24894ef3bd3664de068b44b0",
    ComponentList: listComponentListbody,
    Vpc: "vpc-4b1c",
```



```
        DataCenter: "",
        BillingType: model.GetCreateClusterReqV11BillingTypeEnum().E_12,
        ClusterName: "mrs_HEbK",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 不使用node_groups参数组，创建一个启用“集群高可用”功能的集群，集群版本号为MRS 3.1.0。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    var listActionStagesBootstrapScripts = []model.BootstrapScriptActionStages{
        model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_IN,
        model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_OUT,
    }
    var listNodesBootstrapScripts = []string{
        "master",
    }
    var listActionStagesBootstrapScripts1 = []model.BootstrapScriptActionStages{
        model.GetBootstrapScriptActionStagesEnum().BEFORE_COMPONENT_FIRST_START,
        model.GetBootstrapScriptActionStagesEnum().BEFORE_SCALE_IN,
```



```
}
var listNodesBootstrapScripts1 = []string{
    "master",
    "core",
    "task",
}
parametersBootstrapScripts:= "param1param2"
activeMasterBootstrapScripts:= false
beforeComponentStartBootstrapScripts:= true
startTimeBootstrapScripts:= int64(1667892101)
stateBootstrapScripts:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
parametersBootstrapScripts1:= ""
activeMasterBootstrapScripts1:= true
beforeComponentStartBootstrapScripts1:= false
startTimeBootstrapScripts1:= int64(1667892101)
stateBootstrapScripts1:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
var listBootstrapScriptsbody = []model.BootstrapScript{
    {
        Name: "Modifyosconfig",
        Uri: "s3a: //XXX/modify_os_config.sh",
        Parameters: &parametersBootstrapScripts,
        Nodes: listNodesBootstrapScripts1,
        ActiveMaster: &activeMasterBootstrapScripts,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts,
        StartTime: &startTimeBootstrapScripts,
        State: &stateBootstrapScripts,
        ActionStages: &listActionStagesBootstrapScripts1,
    },
    {
        Name: "Installzepplin",
        Uri: "s3a: //XXX/zeppelin_install.sh",
        Parameters: &parametersBootstrapScripts1,
        Nodes: listNodesBootstrapScripts,
        ActiveMaster: &activeMasterBootstrapScripts1,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts1,
        StartTime: &startTimeBootstrapScripts1,
        State: &stateBootstrapScripts1,
        ActionStages: &listActionStagesBootstrapScripts,
    },
}
var listNodesExecScripts = []string{
    "master",
    "core",
    "task",
}
var listNodesExecScripts1 = []string{
    "master",
    "core",
    "task",
}
parametersExecScripts:= "${mrs_scale_node_num}${mrs_scale_type}xxx"
activeMasterExecScripts:= true
parametersExecScripts1:= "${mrs_scale_node_hostnames}${mrs_scale_node_ips}"
activeMasterExecScripts1:= true
var listExecScriptsAutoScalingPolicy = []model.ScaleScript{
    {
        Name: "before_scale_out",
        Uri: "s3a: //XXX/zeppelin_install.sh",
        Parameters: &parametersExecScripts,
        Nodes: listNodesExecScripts1,
        ActiveMaster: &activeMasterExecScripts,
        FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
        ActionStage: model.GetScaleScriptActionStageEnum().BEFORE_SCALE_OUT,
    },
    {
        Name: "after_scale_out",
        Uri: "s3a: //XXX/storm_rebalance.sh",
```



```
        Parameters: &parametersExecScripts1,
        Nodes: listNodesExecScripts,
        ActiveMaster: &activeMasterExecScripts1,
        FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
        ActionStage: model.GetScaleScriptActionStageEnum().AFTER_SCALE_OUT,
    },
}
comparisonOperatorTrigger:= "GT"
triggerRules := &model.Trigger{
    MetricName: "YARNMemoryAvailablePercentage",
    MetricValue: "70",
    ComparisonOperator: &comparisonOperatorTrigger,
    EvaluationPeriods: int32(10),
}
comparisonOperatorTrigger1:= "LT"
triggerRules1 := &model.Trigger{
    MetricName: "YARNMemoryAvailablePercentage",
    MetricValue: "25",
    ComparisonOperator: &comparisonOperatorTrigger1,
    EvaluationPeriods: int32(10),
}
var listRulesAutoScalingPolicy = []model.Rule{
    {
        Name: "default-expand-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
        CoolDownMinutes: int32(5),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules1,
    },
    {
        Name: "default-shrink-1",
        AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_IN,
        CoolDownMinutes: int32(5),
        ScalingAdjustment: int32(1),
        Trigger: triggerRules,
    },
}
var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
    {
        PeriodType: "daily",
        StartTime: "9: 50",
        EndTime: "10: 20",
        MinCapacity: int32(2),
        MaxCapacity: int32(3),
    },
    {
        PeriodType: "daily",
        StartTime: "10: 20",
        EndTime: "12: 30",
        MinCapacity: int32(0),
        MaxCapacity: int32(2),
    },
}
autoScalingPolicyTaskNodeGroups := &model.AutoScalingPolicy{
    AutoScalingEnable: true,
    MinCapacity: int32(1),
    MaxCapacity: int32(3),
    ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
    Rules: &listRulesAutoScalingPolicy,
    ExecScripts: &listExecScriptsAutoScalingPolicy,
}
var listTaskNodeGroupsbody = []model.TaskNodeGroup{
    {
        NodeNum: int32(2),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        DataVolumeType: model.GetTaskNodeGroupDataVolumeTypeEnum().SATA,
        DataVolumeCount: int32(1),
        DataVolumeSize: int32(600),
        AutoScalingPolicy: autoScalingPolicyTaskNodeGroups,
```



```
    },
  }
  jarPathAddJobs:= "s3a: //bigdata/program/hadoop-mapreduce-examples-2.7.2.jar"
  argumentsAddJobs:= "wordcount"
  inputAddJobs:= "s3a: //bigdata/input/wd_1k/"
  outputAddJobs:= "s3a: //bigdata/ouput/"
  jobLogAddJobs:= "s3a: //bigdata/log/"
  hiveScriptPathAddJobs:= ""
  hqlAddJobs:= ""
  shutdownClusterAddJobs:= true
  fileActionAddJobs:= ""
  var listAddJobsbody = []model.AddJobsReqV11{
    {
      JobType: int32(1),
      JobName: "tenji111",
      JarPath: &jarPathAddJobs,
      Arguments: &argumentsAddJobs,
      Input: &inputAddJobs,
      Output: &outputAddJobs,
      JobLog: &jobLogAddJobs,
      HiveScriptPath: &hiveScriptPathAddJobs,
      Hql: &hqlAddJobs,
      ShutdownCluster: &shutdownClusterAddJobs,
      SubmitJobOnceClusterRun: true,
      FileAction: &fileActionAddJobs,
    },
  }
  var listComponentListbody = []model.ComponentAmbV11{
    {
      ComponentName: "Hadoop",
    },
    {
      ComponentName: "Spark",
    },
    {
      ComponentName: "HBase",
    },
    {
      ComponentName: "Hive",
    },
  }
  logCollectionCreateClusterReqV11:= model.GetCreateClusterReqV11LogCollectionEnum().E_1
  clusterTypeCreateClusterReqV11:= model.GetCreateClusterReqV11ClusterTypeEnum().E_0
  nodePublicCertNameCreateClusterReqV11:= "SSHkey-bba1"
  coreDataVolumeCountCreateClusterReqV11:= int32(2)
  coreDataVolumeSizeCreateClusterReqV11:= int32(600)
  coreDataVolumeTypeCreateClusterReqV11:=
model.GetCreateClusterReqV11CoreDataVolumeTypeEnum().SATA
  masterDataVolumeCountCreateClusterReqV11:=
model.GetCreateClusterReqV11MasterDataVolumeCountEnum().E_1
  masterDataVolumeSizeCreateClusterReqV11:= int32(600)
  masterDataVolumeTypeCreateClusterReqV11:=
model.GetCreateClusterReqV11MasterDataVolumeTypeEnum().SATA
  securityGroupsIdCreateClusterReqV11:= "845bece1-fd22-4b45-7a6e-14338c99ee43"
  coreNodeSizeCreateClusterReqV11:= "s1.xlarge.linux.bigdata"
  masterNodeSizeCreateClusterReqV11:= "s3.2xlarge.2.linux.bigdata"
  coreNodeNumCreateClusterReqV11:= int32(3)
  masterNodeNumCreateClusterReqV11:= int32(2)
  request.Body = &model.CreateClusterReqV11{
    Tags: &listTagsbody,
    LogCollection: &logCollectionCreateClusterReqV11,
    ClusterType: &clusterTypeCreateClusterReqV11,
    SafeMode: model.GetCreateClusterReqV11SafeModeEnum().E_0,
    NodePublicCertName: &nodePublicCertNameCreateClusterReqV11,
    BootstrapScripts: &listBootstrapScriptsbody,
    TaskNodeGroups: &listTaskNodeGroupsbody,
    CoreDataVolumeCount: &coreDataVolumeCountCreateClusterReqV11,
    CoreDataVolumeSize: &coreDataVolumeSizeCreateClusterReqV11,
    CoreDataVolumeType: &coreDataVolumeTypeCreateClusterReqV11,
```



```
        MasterDataVolumeCount: &masterDataVolumeCountCreateClusterReqV11,
        MasterDataVolumeSize: &masterDataVolumeSizeCreateClusterReqV11,
        MasterDataVolumeType: &masterDataVolumeTypeCreateClusterReqV11,
        AddJobs: &listAddJobsbody,
        SecurityGroupsId: &securityGroupsIdCreateClusterReqV11,
        SubnetName: "subnet",
        SubnetId: "815bece0-fd22-4b65-8a6e-15788c99ee43",
        VpcId: "5b7db34d-3534-4a6e-ac94-023cd36aaf74",
        AvailableZoneId: "d573142f24894ef3bd3664de068b44b0",
        ComponentList: listComponentListbody,
        CoreNodeSize: &coreNodeSizeCreateClusterReqV11,
        MasterNodeSize: &masterNodeSizeCreateClusterReqV11,
        Vpc: "vpc1",
        DataCenter: "",
        BillingType: model.GetCreateClusterReqV11BillingTypeEnum().E_12,
        CoreNodeNum: &coreNodeNumCreateClusterReqV11,
        MasterNodeNum: &masterNodeNumCreateClusterReqV11,
        ClusterName: "newcluster",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateClusterRequest{}
    rootVolumeSizeNodeGroups:= "480"
    rootVolumeTypeNodeGroups:= "SATA"
    dataVolumeTypeNodeGroups:= "SATA"
    dataVolumeCountNodeGroups:= int32(1)
    dataVolumeSizeNodeGroups:= int32(600)
```



```
rootVolumeSizeNodeGroups1:= "480"
rootVolumeTypeNodeGroups1:= "SATA"
dataVolumeTypeNodeGroups1:= "SATA"
dataVolumeCountNodeGroups1:= int32(1)
dataVolumeSizeNodeGroups1:= int32(600)
var listNodeGroupsbody = []model.NodeGroupV11{
    {
        GroupName: "master_node_default_group",
        NodeNum: int32(1),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        RootVolumeSize: &rootVolumeSizeNodeGroups,
        RootVolumeType: &rootVolumeTypeNodeGroups,
        DataVolumeType: &dataVolumeTypeNodeGroups,
        DataVolumeCount: &dataVolumeCountNodeGroups,
        DataVolumeSize: &dataVolumeSizeNodeGroups,
    },
    {
        GroupName: "core_node_analysis_group",
        NodeNum: int32(1),
        NodeSize: "s3.xlarge.2.linux.bigdata",
        RootVolumeSize: &rootVolumeSizeNodeGroups1,
        RootVolumeType: &rootVolumeTypeNodeGroups1,
        DataVolumeType: &dataVolumeTypeNodeGroups1,
        DataVolumeCount: &dataVolumeCountNodeGroups1,
        DataVolumeSize: &dataVolumeSizeNodeGroups1,
    },
}
var listTagsbody = []model.Tag{
    {
        Key: "key1",
        Value: "value1",
    },
    {
        Key: "key2",
        Value: "value2",
    },
}
var listActionStagesBootstrapScripts = []model.BootstrapScriptActionStages{
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_IN,
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_OUT,
}
var listNodesBootstrapScripts = []string{
    "master",
}
var listActionStagesBootstrapScripts1 = []model.BootstrapScriptActionStages{
    model.GetBootstrapScriptActionStagesEnum().BEFORE_COMPONENT_FIRST_START,
    model.GetBootstrapScriptActionStagesEnum().BEFORE_SCALE_IN,
}
var listNodesBootstrapScripts1 = []string{
    "master",
    "core",
    "task",
}
parametersBootstrapScripts:= "param1 param2"
activeMasterBootstrapScripts:= false
beforeComponentStartBootstrapScripts:= true
startTimeBootstrapScripts:= int64(1667892101)
stateBootstrapScripts:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
parametersBootstrapScripts1:= ""
activeMasterBootstrapScripts1:= true
beforeComponentStartBootstrapScripts1:= false
startTimeBootstrapScripts1:= int64(1667892101)
stateBootstrapScripts1:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
var listBootstrapScriptsbody = []model.BootstrapScript{
    {
        Name: "Modify os config",
        Uri: "s3a://XXX/modify_os_config.sh",
        Parameters: &parametersBootstrapScripts,
        Nodes: listNodesBootstrapScripts1,
```



```
        ActiveMaster: &activeMasterBootstrapScripts,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts,
        StartTime: &startTimeBootstrapScripts,
        State: &stateBootstrapScripts,
        ActionStages: &listActionStagesBootstrapScripts1,
    },
    {
        Name: "Install zepplin",
        Uri: "s3a://XXX/zeppelin_install.sh",
        Parameters: &parametersBootstrapScripts1,
        Nodes: listNodesBootstrapScripts,
        ActiveMaster: &activeMasterBootstrapScripts1,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts1,
        StartTime: &startTimeBootstrapScripts1,
        State: &stateBootstrapScripts1,
        ActionStages: &listActionStagesBootstrapScripts,
    },
}
jarPathAddJobs:= "s3a://bigdata/program/hadoop-mapreduce-examples-2.7.2.jar"
argumentsAddJobs:= "wordcount"
inputAddJobs:= "s3a://bigdata/input/wd_1k/"
outputAddJobs:= "s3a://bigdata/ouput/"
jobLogAddJobs:= "s3a://bigdata/log/"
hiveScriptPathAddJobs:= ""
hqlAddJobs:= ""
shutdownClusterAddJobs:= true
fileActionAddJobs:= ""
var listAddJobsbody = []model.AddJobsReqV11{
    {
        JobType: int32(1),
        JobName: "tenji111",
        JarPath: &jarPathAddJobs,
        Arguments: &argumentsAddJobs,
        Input: &inputAddJobs,
        Output: &outputAddJobs,
        JobLog: &jobLogAddJobs,
        HiveScriptPath: &hiveScriptPathAddJobs,
        Hql: &hqlAddJobs,
        ShutdownCluster: &shutdownClusterAddJobs,
        SubmitJobOnceClusterRun: true,
        FileAction: &fileActionAddJobs,
    },
}
var listComponentListbody = []model.ComponentAmbV11{
    {
        ComponentName: "Hadoop",
    },
    {
        ComponentName: "Spark",
    },
    {
        ComponentName: "HBase",
    },
    {
        ComponentName: "Hive",
    },
    {
        ComponentName: "Presto",
    },
    {
        ComponentName: "Tez",
    },
    {
        ComponentName: "Hue",
    },
    {
        ComponentName: "Loader",
```



```
        },
        {
            ComponentName: "Flink",
        },
    }
    loginModeCreateClusterReqV11:= model.GetCreateClusterReqV11LoginModeEnum().E_1
    enterpriseProjectIdCreateClusterReqV11:= "0"
    logCollectionCreateClusterReqV11:= model.GetCreateClusterReqV11LogCollectionEnum().E_1
    clusterTypeCreateClusterReqV11:= model.GetCreateClusterReqV11ClusterTypeEnum().E_0
    clusterMasterSecretCreateClusterReqV11:= ""
    clusterAdminSecretCreateClusterReqV11:= ""
    securityGroupsIdCreateClusterReqV11:= "4820eace-66ad-4f2c-8d46-cf340e3029dd"
    request.Body = &model.CreateClusterReqV11{
        NodeGroups: &listNodeGroupsbody,
        LoginMode: &loginModeCreateClusterReqV11,
        Tags: &listTagsbody,
        EnterpriseProjectId: &enterpriseProjectIdCreateClusterReqV11,
        LogCollection: &logCollectionCreateClusterReqV11,
        ClusterType: &clusterTypeCreateClusterReqV11,
        SafeMode: model.GetCreateClusterReqV11SafeModeEnum().E_0,
        ClusterMasterSecret: &clusterMasterSecretCreateClusterReqV11,
        ClusterAdminSecret: &clusterAdminSecretCreateClusterReqV11,
        BootstrapScripts: &listBootstrapScriptsbody,
        AddJobs: &listAddJobsbody,
        SecurityGroupsId: &securityGroupsIdCreateClusterReqV11,
        SubnetName: "subnet-4b44",
        SubnetId: "67984709-e15e-4e86-9886-d76712d4e00a",
        VpcId: "4a365717-67be-4f33-80c5-98e98a813af8",
        AvailableZoneId: "d573142f24894ef3bd3664de068b44b0",
        ComponentList: listComponentListbody,
        Vpc: "vpc-4b1c",
        DataCenter: "",
        BillingType: model.GetCreateClusterReqV11BillingTypeEnum().E_12,
        ClusterName: "mrs_HEbK",
        ClusterVersion: "MRS 3.1.0",
    }
    response, err := client.CreateCluster(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 不使用node_groups参数组，创建一个关闭“集群高可用”功能、最小规格的集群，集群版本号为MRS 3.1.0。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := mrs.NewMrsClient(
    mrs.MrsClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateClusterRequest{}
var listTagsbody = []model.Tag{
    {
        Key: "key1",
        Value: "value1",
    },
    {
        Key: "key2",
        Value: "value2",
    },
}
var listActionStagesBootstrapScripts = []model.BootstrapScriptActionStages{
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_IN,
    model.GetBootstrapScriptActionStagesEnum().AFTER_SCALE_OUT,
}
var listNodesBootstrapScripts = []string{
    "master",
}
parametersBootstrapScripts:= ""
activeMasterBootstrapScripts:= false
beforeComponentStartBootstrapScripts:= false
startTimeBootstrapScripts:= int64(1667892101)
stateBootstrapScripts:= model.GetBootstrapScriptStateEnum().IN_PROGRESS
var listBootstrapScriptsbody = []model.BootstrapScript{
    {
        Name: "Install zepplin",
        Uri: "s3a://XXX/zeppelin_install.sh",
        Parameters: &parametersBootstrapScripts,
        Nodes: listNodesBootstrapScripts,
        ActiveMaster: &activeMasterBootstrapScripts,
        FailAction: model.GetBootstrapScriptFailActionEnum().CONTINUE,
        BeforeComponentStart: &beforeComponentStartBootstrapScripts,
        StartTime: &startTimeBootstrapScripts,
        State: &stateBootstrapScripts,
        ActionStages: &listActionStagesBootstrapScripts,
    },
}
jarPathAddJobs:= "s3a://bigdata/program/hadoop-mapreduce-examples-XXX.jar"
argumentsAddJobs:= "wordcount"
inputAddJobs:= "s3a://bigdata/input/wd_1k/"
outputAddJobs:= "s3a://bigdata/ouput/"
jobLogAddJobs:= "s3a://bigdata/log/"
hiveScriptPathAddJobs:= ""
hqlAddJobs:= ""
shutdownClusterAddJobs:= false
fileActionAddJobs:= ""
var listAddJobsbody = []model.AddJobsReqV11{
    {
        JobType: int32(1),
        JobName: "tenji111",
        JarPath: &jarPathAddJobs,
        Arguments: &argumentsAddJobs,
        Input: &inputAddJobs,
        Output: &outputAddJobs,
        JobLog: &jobLogAddJobs,
        HiveScriptPath: &hiveScriptPathAddJobs,
        Hql: &hqlAddJobs,
        ShutdownCluster: &shutdownClusterAddJobs,
        SubmitJobOnceClusterRun: true,
```



```
            FileAction: &fileActionAddJobs,
        },
    }
    var listComponentListbody = []model.ComponentAmbV11{
        {
            ComponentName: "Hadoop",
        },
        {
            ComponentName: "Spark",
        },
        {
            ComponentName: "HBase",
        },
        {
            ComponentName: "Hive",
        },
        {
            ComponentName: "Presto",
        },
        {
            ComponentName: "Tez",
        },
        {
            ComponentName: "Hue",
        },
        {
            ComponentName: "Loader",
        },
        {
            ComponentName: "Flink",
        },
    }
    loginModeCreateClusterReqV11:= model.GetCreateClusterReqV11LoginModeEnum().E_1
    enterpriseProjectIdCreateClusterReqV11:= "0"
    logCollectionCreateClusterReqV11:= model.GetCreateClusterReqV11LogCollectionEnum().E_1
    clusterTypeCreateClusterReqV11:= model.GetCreateClusterReqV11ClusterTypeEnum().E_0
    clusterAdminSecretCreateClusterReqV11:= "******"
    nodePublicCertNameCreateClusterReqV11:= "SSHkey-bba1"
    coreDataVolumeCountCreateClusterReqV11:= int32(1)
    coreDataVolumeSizeCreateClusterReqV11:= int32(600)
    coreDataVolumeTypeCreateClusterReqV11:=
model.GetCreateClusterReqV11CoreDataVolumeTypeEnum().SATA
    masterDataVolumeCountCreateClusterReqV11:=
model.GetCreateClusterReqV11MasterDataVolumeCountEnum().E_1
    masterDataVolumeSizeCreateClusterReqV11:= int32(600)
    masterDataVolumeTypeCreateClusterReqV11:=
model.GetCreateClusterReqV11MasterDataVolumeTypeEnum().SATA
    securityGroupsIdCreateClusterReqV11:= ""
    coreNodeSizeCreateClusterReqV11:= "s1.xlarge.linux.bigdata"
    masterNodeSizeCreateClusterReqV11:= "s3.2xlarge.2.linux.bigdata"
    coreNodeNumCreateClusterReqV11:= int32(1)
    masterNodeNumCreateClusterReqV11:= int32(1)
    request.Body = &model.CreateClusterReqV11{
        LoginMode: &loginModeCreateClusterReqV11,
        Tags: &listTagsbody,
        EnterpriseProjectId: &enterpriseProjectIdCreateClusterReqV11,
        LogCollection: &logCollectionCreateClusterReqV11,
        ClusterType: &clusterTypeCreateClusterReqV11,
        SafeMode: model.GetCreateClusterReqV11SafeModeEnum().E_0,
        ClusterAdminSecret: &clusterAdminSecretCreateClusterReqV11,
        NodePublicCertName: &nodePublicCertNameCreateClusterReqV11,
        BootstrapScripts: &listBootstrapScriptsbody,
        CoreDataVolumeCount: &coreDataVolumeCountCreateClusterReqV11,
        CoreDataVolumeSize: &coreDataVolumeSizeCreateClusterReqV11,
        CoreDataVolumeType: &coreDataVolumeTypeCreateClusterReqV11,
        MasterDataVolumeCount: &masterDataVolumeCountCreateClusterReqV11,
        MasterDataVolumeSize: &masterDataVolumeSizeCreateClusterReqV11,
        MasterDataVolumeType: &masterDataVolumeTypeCreateClusterReqV11,
        AddJobs: &listAddJobsbody,
```



```
        SecurityGroupsId: &securityGroupsIdCreateClusterReqV11,
        SubnetName: "subnet",
        SubnetId: "815bece0-fd22-4b65-8a6e-15788c99ee43",
        VpcId: "5b7db34d-3534-4a6e-ac94-023cd36aaf74",
        AvailableZoneId: "d573142f24894ef3bd3664de068b44b0",
        ComponentList: listComponentListbody,
        CoreNodeSize: &coreNodeSizeCreateClusterReqV11,
        MasterNodeSize: &masterNodeSizeCreateClusterReqV11,
        Vpc: "vpc1",
        DataCenter: "",
        BillingType: model.GetCreateClusterReqV11BillingTypeEnum().E_12,
        CoreNodeNum: &coreNodeNumCreateClusterReqV11,
        MasterNodeNum: &masterNodeNumCreateClusterReqV11,
        ClusterName: "newcluster",
        ClusterVersion: "MRS 3.1.0",
    }
    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。

错误码

请参见**错误码**。

7.1.2 调整集群节点

功能介绍

创建集群后，扩容/缩容集群Core节点或者Task节点。MRS集群创建成功后不支持调整Master节点数量，即不支持扩缩容Master节点。该接口不兼容Sahara。处于running状态的集群才允许扩容/缩容，其他状态则不允许扩容/缩容。 集群状态和集群ID可参考**查询集群列表**接口获取。 本章节的接口只支持流式集群、分析集群和混合集群，不支持自定义集群。

接口约束

无

调用方法

请参见**如何调用API**。

URI

PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

表 7-15 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-16 请求 Body 参数

参数	是否必选	参数类型	描述
service_id	否	String	**参数解释：** 服务ID，为扩展接口，预留此参数。用户不需要配置。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
plan_id	否	String	**参数解释：** 套餐ID，为扩展接口，预留此参数。用户不需要配置。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	是	**ClusterScalingParams** object	**参数解释：** 请求参数集合。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
previous_values	否	Map<String,String>	**参数解释：** 扩展接口，预留此参数。用户不需要配置。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-17 ClusterScalingParams

参数	是否必选	参数类型	描述
order_id	否	String	**参数解释：** 扩容/缩容时系统获取的订单号，用户不需要配置。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
scale_type	是	String	**参数解释：** 范围操作类型。 **约束限制：** 不涉及 **取值范围：** ● scale_in：缩容 ● scale_out：扩容 **默认取值：** 不涉及
node_id	是	String	**参数解释：** 扩容/缩容时新增或者减少节点的ID标识,参数值固定为node_orderadd。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
node_group	否	String	**参数解释：** 扩容或缩容的节点组。 **约束限制：** 不涉及 **取值范围：** ● core_node_default_group：表示Core节点组。 ● task_node_default_group：表示Task节点组。 **默认取值：** core_node_default_group
skip_bootstrap_scripts	否	String	**参数解释：** 是否跳过引导操作。仅在创建集群时配置了引导操作且扩容时有意义，表示扩容时是否在新增节点上执行创建集群时指定的引导操作。 **约束限制：** 不涉及 **取值范围：** ● true：跳过引导操作。 ● false：执行引导操作。 **默认取值：** false
scale_without_start	否	Boolean	**参数解释：** 扩容后是否启动扩容节点上的组件。 **约束限制：** 不涉及 **取值范围：** ● true：扩容后不启动组件。 ● false：扩容后启动组件。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
server_ids	否	Array of strings	**参数解释：** 缩容Task节点时指定待删除Task节点的ID列表。 **约束限制：** ● 当scale_type为扩容时，该参数不生效。 ● 当scale_type为缩容且该参数不为空时，删除指定的Task节点。 ● 当scale_type为缩容且server_ids为空时，按照系统规则自动选择删除Task节点。 **取值范围：** 不涉及 **默认取值：** 不涉及
instances	是	Integer	**参数解释：** 扩容或缩容的节点数。 **约束限制：** ● 扩容时的最大节点数为（500 - 集群Core/Task节点数）。例如，当前集群Core节点数为3，此处扩容的节点数必须小于等于497。 ● Core和Task节点总数最大值为500，如果用户需要的Core/Task节点数大于500，可以联系技术支持人员或者调用后台接口修改数据库。 -缩容时Core节点数大于3或者Task节点数大于0可以进行节点删除。例如，当前集群Core节点和Task节点数均为5，Core节点可缩容的节点数为2（5减去3），Task节点可缩容节点数为小于等于5。 **取值范围：** ≥1 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
task_node_info	否	TaskNodeInfo object	**参数解释：** Task节点规格。 **约束限制：** • 当Task节点个数为0时，使用该字段指定Task节点的规格。 • 当Task节点个数大于0时，不能使用该字段。该字段为空时，不填即可，字段内含必填参数不可使用会传空值对象的置空按钮。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-18 TaskNodeInfo

参数	是否必选	参数类型	描述
node_size	是	String	**参数解释：** Task节点的实例规格，例如：{ECS_FLAVOR_NAME}.linux.bigdata，{ECS_FLAVOR_NAME}可以为c3.4xlare.2等在MRS购买页可见的云服务器规格。实例规格详细说明请参见**MRS所使用的弹性云服务器规格**和**MRS所使用的裸金属服务器规格**。该参数建议从MRS控制台的集群创建页面获取对应区域对应版本所支持的规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
data_volume_type	是	String	**参数解释：** Task节点数据磁盘存储类别，目前支持SATA、SAS和SSD等。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO ● GPSSD：通用型SSD **默认取值：** 不涉及
data_volume_count	是	Integer	**参数解释：** Task节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 0-20 **默认取值：** 不涉及
data_volume_size	是	Integer	**参数解释：** Task节点数据磁盘存储大小。传值只需填数字，不需要带单位GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及

响应参数

状态码： 200

表 7-19 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 操作结果。 **约束限制：** ● succeeded：操作成功。 ● 操作失败时返回的错误码信息如**错误码**所示。 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- 在MRS集群中扩容1个core节点。

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_out",
    "node_id" : "node_orderadd",
    "node_group" : "core_node_default_group",
    "instances" : "1",
    "skip_bootstrap_scripts" : false,
    "scale_without_start" : false
  },
  "previous_values" : { }
}
```

- 当Task节点个数大于零时，在MRS集群中扩容1个Task节点。

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_out",
    "node_id" : "node_orderadd",
    "node_group" : "task_node_default_group",
    "instances" : "1",
    "skip_bootstrap_scripts" : false,
    "scale_without_start" : false
  },
  "previous_values" : { }
}
```

- 当Task节点个数等于零时，在MRS集群中扩容1个规格为s3.xlarge.2.linux.bigdata的Task节点。

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
```

```
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_out",
    "node_id" : "node_orderadd",
    "node_group" : "task_node_default_group",
    "task_node_info" : {
      "node_size" : "s3.xlarge.2.linux.bigdata",
      "data_volume_type" : "SATA",
      "data_volume_count" : 2,
      "data_volume_size" : 600
    },
    "instances" : "1",
    "scale_without_start" : false
  },
  "previous_values" : { }
}
```

- 在MRS集群中缩容1个Core节点。

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_in",
    "node_id" : "node_orderadd",
    "node_group" : "core_node_default_group",
    "instances" : "1"
  },
  "previous_values" : { }
}
```

- 在MRS集群中缩容1个Task节点

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_in",
    "node_id" : "node_orderadd",
    "node_group" : "task_node_default_group",
    "instances" : "1"
  },
  "previous_values" : { }
}
```

- 在MRS集群中缩容指定的Task节点。

```
PUT /v1.1/{project_id}/cluster_infos/{cluster_id}

{
  "service_id" : "",
  "plan_id" : "",
  "parameters" : {
    "order_id" : "",
    "scale_type" : "scale_in",
    "node_id" : "node_orderadd",
    "node_group" : "task_node_default_group",
    "instances" : "2",
    "server_ids" : [ "c9573435-7814-4b2c-9131-ad78b814414c",
"a4951009-6a0f-4e7b-9c81-9d4bd1f8c537" ]
  },
  "previous_values" : { }
}
```


响应示例

状态码： 200

扩容/缩容集群Core节点或者Task节点成功。

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 在MRS集群中扩容1个core节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_out"))
            .withNodeId("node_orderadd")
            .withNodeGroup("core_node_default_group")
            .withSkipBootstrapScripts("false")
            .withScaleWithoutStart(false)
            .withInstances(1);
        body.withParameters(parametersbody);
        body.withPlanId("");
        body.withServiceId("");
```

```
        request.withBody(body);
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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());
        }
    }
}
```

- 当Task节点个数大于零时，在MRS集群中扩容1个Task节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_out"))
            .withNodeId("node_orderadd")
            .withNodeGroup("task_node_default_group")
            .withSkipBootstrapScripts("false")
            .withScaleWithoutStart(false)
            .withInstances(1);
        body.withParameters(parametersbody);
        body.withPlanId("");
        body.withServiceId("");
        request.withBody(body);
```



```
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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());
        }
    }
}
```

- 当Task节点个数等于零时，在MRS集群中扩容1个规格为s3.xlarge.2.linux.bigdata的Task节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        TaskNodeInfo taskNodeInfoParameters = new TaskNodeInfo();
        taskNodeInfoParameters.withNodeSize("s3.xlarge.2.linux.bigdata")
            .withDataVolumeType(TaskNodeInfo.DataVolumeTypeEnum.fromValue("SATA"))
            .withDataVolumeCount(2)
            .withDataVolumeSize(600);
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_out"))
            .withNodeId("node_orderadd")
            .withNodeGroup("task_node_default_group")
            .withScaleWithoutStart(false)
```



```
            .withInstances(1)
            .withTaskNodeInfo(taskNodeInfoParameters);
        body.withParameters(parametersbody);
        body.withPlanId("");
        body.withServiceId("");
        request.withBody(body);
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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());
        }
    }
}
```

- 在MRS集群中缩容1个Core节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_in"))
            .withNodeId("node_orderadd")
            .withNodeGroup("core_node_default_group")
            .withInstances(1);
        body.withParameters(parametersbody);
```



```
        body.withPlanId("");
        body.withServiceId("");
        request.withBody(body);
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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());
        }
    }
}
```

- 在MRS集群中缩容1个Task节点

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_in"))
            .withNodeId("node_orderadd")
            .withNodeGroup("task_node_default_group")
            .withInstances(1);
        body.withParameters(parametersbody);
        body.withPlanId("");
        body.withServiceId("");
        request.withBody(body);
```



```
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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());
        }
    }
}
```

- 在MRS集群中缩容指定的Task节点。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class UpdateClusterScalingSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateClusterScalingRequest request = new UpdateClusterScalingRequest();
        request.withClusterId("{cluster_id}");
        ClusterScalingReq body = new ClusterScalingReq();
        List<String> listParametersServerIds = new ArrayList<>();
        listParametersServerIds.add("c9573435-7814-4b2c-9131-ad78b814414c");
        listParametersServerIds.add("a4951009-6a0f-4e7b-9c81-9d4bd1f8c537");
        ClusterScalingParams parametersbody = new ClusterScalingParams();
        parametersbody.withOrderId("")
            .withScaleType(ClusterScalingParams.ScaleTypeEnum.fromValue("scale_in"))
            .withNodeId("node_orderadd")
            .withNodeGroup("task_node_default_group")
            .withServerIds(listParametersServerIds)
            .withInstances(2);
        body.withParameters(parametersbody);
```



```
        body.withPlanId("");
        body.withServiceId("");
        request.withBody(body);
        try {
            UpdateClusterScalingResponse response = client.updateClusterScaling(request);
            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

- 在MRS集群中扩容1个core节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_out",
            node_id="node_orderadd",
            node_group="core_node_default_group",
            skip_bootstrap_scripts="false",
            scale_without_start=False,
            instances=1
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
        print(response)
    except exceptions.ClientRequestException as e:
```



```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 当Task节点个数大于零时，在MRS集群中扩容1个Task节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_out",
            node_id="node_orderadd",
            node_group="task_node_default_group",
            skip_bootstrap_scripts="false",
            scale_without_start=False,
            instances=1
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 当Task节点个数等于零时，在MRS集群中扩容1个规格为s3.xlarge.2.linux.bigdata的Task节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```



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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        taskNodeInfoParameters = TaskNodeInfo(
            node_size="s3.xlarge.2.linux.bigdata",
            data_volume_type="SATA",
            data_volume_count=2,
            data_volume_size=600
        )
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_out",
            node_id="node_orderadd",
            node_group="task_node_default_group",
            scale_without_start=False,
            instances=1,
            task_node_info=taskNodeInfoParameters
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 在MRS集群中缩容1个Core节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

    client = MrsClient.new_builder() \
```



```
        .with_credentials(credentials) \
        .with_region(MrsRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_in",
            node_id="node_orderadd",
            node_group="core_node_default_group",
            instances=1
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 在MRS集群中缩容1个Task节点

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_in",
            node_id="node_orderadd",
            node_group="task_node_default_group",
            instances=1
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
```



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

- 在MRS集群中缩容指定的Task节点。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateClusterScalingRequest()
        request.cluster_id = "{cluster_id}"
        listServerIdsParameters = [
            "c9573435-7814-4b2c-9131-ad78b814414c",
            "a4951009-6a0f-4e7b-9c81-9d4bd1f8c537"
        ]
        parametersbody = ClusterScalingParams(
            order_id="",
            scale_type="scale_in",
            node_id="node_orderadd",
            node_group="task_node_default_group",
            server_ids=listServerIdsParameters,
            instances=2
        )
        request.body = ClusterScalingReq(
            parameters=parametersbody,
            plan_id="",
            service_id=""
        )
        response = client.update_cluster_scaling(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 在MRS集群中扩容1个core节点。

```
package main

import (
```

```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterScalingRequest{}
    request.ClusterId = "{cluster_id}"
    orderIdParameters:= ""
    nodeGroupParameters:= "core_node_default_group"
    skipBootstrapScriptsParameters:= "false"
    scaleWithoutStartParameters:= false
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_OUT,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        SkipBootstrapScripts: &skipBootstrapScriptsParameters,
        ScaleWithoutStart: &scaleWithoutStartParameters,
        Instances: int32(1),
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 当Task节点个数大于零时，在MRS集群中扩容1个Task节点。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)
```



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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterScalingRequest{}
    request.ClusterId = "{cluster_id}"
    orderIdParameters:= ""
    nodeGroupParameters:= "task_node_default_group"
    skipBootstrapScriptsParameters:= "false"
    scaleWithoutStartParameters:= false
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_OUT,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        SkipBootstrapScripts: &skipBootstrapScriptsParameters,
        ScaleWithoutStart: &scaleWithoutStartParameters,
        Instances: int32(1),
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 当Task节点个数等于零时，在MRS集群中扩容1个规格为s3.xlarge.2.linux.bigdata的Task节点。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterScalingRequest{}
    request.ClusterId = "{cluster_id}"
    taskNodeInfoParameters := &model.TaskNodeInfo{
        NodeSize: "s3.xlarge.2.linux.bigdata",
        DataVolumeType: model.GetTaskNodeInfoDataVolumeTypeEnum().SATA,
        DataVolumeCount: int32(2),
        DataVolumeSize: int32(600),
    }
    orderIdParameters:= ""
    nodeGroupParameters:= "task_node_default_group"
    scaleWithoutStartParameters:= false
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_OUT,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        ScaleWithoutStart: &scaleWithoutStartParameters,
        Instances: int32(1),
        TaskNodeInfo: taskNodeInfoParameters,
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 在MRS集群中缩容1个Core节点。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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



```
    // In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterScalingRequest{}
    request.ClusterId = "{cluster_id}"
    orderIdParameters:= ""
    nodeGroupParameters:= "core_node_default_group"
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_IN,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        Instances: int32(1),
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 在MRS集群中缩容1个Task节点

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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



```
        WithProjectId(projectId).
        Build()

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateClusterScalingRequest{}
    request.ClusterId = "{cluster_id}"
    orderIdParameters:= ""
    nodeGroupParameters:= "task_node_default_group"
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_IN,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        Instances: int32(1),
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 在MRS集群中缩容指定的Task节点。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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



```
    request.ClusterId = "{cluster_id}"
    var listServerIdsParameters = []string{
        "c9573435-7814-4b2c-9131-ad78b814414c",
        "a4951009-6a0f-4e7b-9c81-9d4bd1f8c537",
    }
    orderIdParameters:= ""
    nodeGroupParameters:= "task_node_default_group"
    parametersbody := &model.ClusterScalingParams{
        OrderId: &orderIdParameters,
        ScaleType: model.GetClusterScalingParamsScaleTypeEnum().SCALE_IN,
        NodeId: "node_orderadd",
        NodeGroup: &nodeGroupParameters,
        ServerIds: &listServerIdsParameters,
        Instances: int32(2),
    }
    planIdClusterScalingReq:= ""
    serviceIdClusterScalingReq:= ""
    request.Body = &model.ClusterScalingReq{
        Parameters: parametersbody,
        PlanId: &planIdClusterScalingReq,
        ServiceId: &serviceIdClusterScalingReq,
    }
    response, err := client.UpdateClusterScaling(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	扩容/缩容集群Core节点或者Task节点成功。

错误码

请参见**错误码**。

7.1.3 查询集群列表

功能介绍

查看用户创建的集群列表信息。该接口不兼容Sahara。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/cluster_infos

表 7-20 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 7-21 Query 参数

参数	是否必选	参数类型	描述
tags	否	String	**参数解释：** 标签列表。可以通过集群的标签来搜索指定标签的集群，当指定多个tag进行查询时，标签之间是与的关系。 **约束限制：** 不涉及 **取值范围：** • tags参数的格式为tags=k1*v1,k2*v2,k3*v3 • 当标签的value为空时，格式为tags=k1,k2,k3*v3 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
pageSize	否	String	**参数解释：** 分页查询每页返回的最大集群数量。 **约束限制：** 不涉及 **取值范围：** 1-2147483646 **默认取值：** 10
currentPage	否	String	**参数解释：** 当前查询页码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 1
clusterName	否	String	**参数解释：** 集群名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
clusterState	否	String	**参数解释：** 根据集群状态查询集群列表。 **约束限制：** 不涉及 **取值范围：** ● existing：查询现有集群列表，包括除“已删除”、包周期集群的“订单处理中”和“准备中”状态外的所有集群。 ● history：查询历史集群列表，包括所有“已删除”、删除集群失败、集群删除虚拟机失败、删除集群更新数据库失败等状态的集群。 ● starting：查询启动中的集群列表。 ● running：查询运行中的集群列表。 ● terminated：查询已删除的集群列表。 ● failed：查询失败的集群列表。 ● abnormal：查询异常的集群列表。 ● terminating：查询删除中的集群列表。 ● frozen：查询已冻结的集群列表。 ● scaling-out：查询扩容中的集群列表。 ● scaling-in：查询缩容中的集群列表。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
enterpriseProjectId	否	String	**参数解释：** 通过企业项目ID来搜索指定项目的集群。获取方式请参见《企业管理API参考》的“查询企业项目列表”响应消息表“enterprise_project字段数据结构说明”的“id”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 0

请求参数

无

响应参数

状态码： 200

表 7-22 响应 Body 参数

参数	参数类型	描述
clusterTotal	Integer	**参数解释：** 集群列表总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
clusters	Array of **Cluster** objects	**参数解释：** 集群参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-23 Cluster

参数	参数类型	描述
clusterId	String	**参数解释：** 集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
clusterName	String	**参数解释：** 集群名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
totalNodeNum	String	**参数解释：** 集群部署的节点总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
clusterState	String	**参数解释：** 集群状态（默认为英文，当X-Language请求头被设置为zh-cn，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** ● starting：启动中。 ● running：运行中。 ● terminated：已删除。 ● failed：失败。 ● abnormal：异常。 ● terminating：删除中。 ● frozen：已冻结。 ● scaling-out：扩容中。 ● scaling-in：缩容中。 **默认取值：** 不涉及

参数	参数类型	描述
stageDesc	String	**参数解释：** 集群进度描述。（默认为英文，当X-Language请求头被设置为zh-cn时，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** 安装集群进度包括： ● Verifying cluster parameters：校验集群参数中 ● Applying for cluster resources：申请集群资源中 ● Creating VM：创建虚拟机中 ● Initializing VM：初始化虚拟机中 ● Installing MRS Manager：安装MRS Manager中 ● Deploying cluster：部署集群中 ● Cluster installation failed：集群安装失败 扩容集群进度包括： ● Preparing for cluster expansion：准备扩容中 ● Creating VM：创建虚拟机中 ● Initializing VM：初始化虚拟机中 ● Adding node to the cluster：节点加入集群中 ● Cluster expansion failed：集群扩容失败 缩容集群进度包括： ● Preparing for cluster shrink：正在准备缩容 ● Decommissioning instance：实例退服中 ● Deleting VM：删除虚拟机中 ● Deleting node from the cluster：从集群删除节点中 ● Cluster shrink failed：集群缩容失败 集群安装、扩容、缩容失败，stageDesc会显示失败的原因。 **默认取值：** 不涉及

参数	参数类型	描述
createAt	String	**参数解释：** 集群创建时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
updateAt	String	**参数解释：** 集群更新时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
chargingStartTime	String	**参数解释：** 开始计费时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
billingType	String	**参数解释：** 集群计费模式。 **约束限制：** 不涉及 **取值范围：** ● 包周期计费 ● 按需计费 **默认取值：** 不涉及

参数	参数类型	描述
dataCenter	String	**参数解释：** 集群工作区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpc	String	**参数解释：** VPC名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpcId	String	**参数解释：** VPC ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
duration	String	**参数解释：** 集群购买时长。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
fee	String	**参数解释：** 创建集群所需费用，系统自动计算。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
hadoopVersion	String	**参数解释：** Hadoop组件版本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentList	Array of **ComponentAmb** objects	**参数解释：** 组件列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
externalIp	String	**参数解释：** 公网IP地址。用于集群外访问集群主master进程所开放IP的地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
externalAlternateIp	String	**参数解释：** 公网备用IP地址。用于集群外访问集群备master进程所使用的IP地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
internalIp	String	**参数解释：** 内网IP地址。用于集群内访问集群master进程所使用IP的地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
deploymentId	String	**参数解释：** 集群部署ID。本集群的任务部署所对应的ID，用于云服务查询集群部署任务详情。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
remark	String	**参数解释：** 集群备注信息。当前默认都为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
orderId	String	**参数解释：** 创建集群的订单号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azId	String	**参数解释：** 可用区域ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azName	String	**参数解释：** 可用区域名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azCode	String	**参数解释：** 可用区域英文名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
availabilityZoneId	String	**参数解释：** 可用区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
instanceId	String	**参数解释：** 实例ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vnc	String	**参数解释：** 远程登录弹性云服务器的URI地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenantId	String	**参数解释：** 项目编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
volumeSize	Integer	**参数解释：** 磁盘存储空间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
volumeType	String	**参数解释：** 磁盘类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
subnetId	String	**参数解释：** 子网ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
subnetName	String	**参数解释：** 子网名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
securityGroupsId	String	**参数解释：** 安全组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
slaveSecurityGroupsId	String	**参数解释：** 非Master节点的安全组id，当前一个MRS集群只会使用一个安全组，所以该字段已经废弃，从兼容性考虑，该字段会返回和securityGroupsId同样的值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
bootstrapScripts	Array of **BootstrapScript** objects	**参数解释：** 配置引导操作脚本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
safeMode	Integer	**参数解释：** MRS集群运行模式。 **约束限制：** 不涉及 **取值范围：** ● 0：普通集群 ● 1：安全集群 **默认取值：** 不涉及

参数	参数类型	描述
clusterVersion	String	**参数解释：** 集群版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
nodePublicCertName	String	**参数解释：** 密钥文件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeIp	String	**参数解释：** Master节点IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
privateIpFirst	String	**参数解释：** 首选私有IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
errorInfo	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	String	**参数解释：** 标签信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeNum	String	**参数解释：** 集群部署的Master节点数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeNum	String	**参数解释：** 集群部署的Core节点数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
masterNodeSize	String	**参数解释：** Master节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeSize	String	**参数解释：** Core节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeProductId	String	**参数解释：** Master节点产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeSpecId	String	**参数解释：** Master节点规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
coreNodeProductId	String	**参数解释：** Core节点产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeSpecId	String	**参数解释：** Core节点规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterDataVolumeType	String	**参数解释：** Master节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterDataVolumeSize	Integer	**参数解释：** Master节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。单位为GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及

参数	参数类型	描述
masterDataVolumeCount	Integer	**参数解释：** Master节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 只能为1。 **默认取值：** 不涉及
coreDataVolumeType	String	**参数解释：** Core节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreDataVolumeSize	Integer	**参数解释：** Core节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。单位为GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及
coreDataVolumeCount	Integer	**参数解释：** Core节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 1-10 **默认取值：** 不涉及

参数	参数类型	描述
enterpriseProjectId	String	**参数解释：** 企业项目ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
isMrsManagerFinish	Boolean	**参数解释：** 表示集群创建过程中，MRS Manager是否安装完成。 **约束限制：** 不涉及 **取值范围：** ● true：安装完成。 ● false：安装未完成。 **默认取值：** 不涉及
clusterType	Integer	**参数解释：** 集群类型。 **约束限制：** 不涉及 **取值范围：** ● 0：分析集群。 ● 1：流式集群。 ● 2：混合集群。 ● 3：自定义集群。 ● 4：离线集群。 **默认取值：** 不涉及

参数	参数类型	描述
logCollection	Integer	**参数解释：** 集群安装失败时，是否搜集日志。 **约束限制：** 不涉及 **取值范围：** ● 0：不收集。 ● 1：收集。 **默认取值：** 不涉及
periodType	Integer	**参数解释：** 区分包周期，集群是包年还是包月。 **约束限制：** 不涉及 **取值范围：** ● 0：包月。 ● 1：包年。 **默认取值：** 不涉及
scale	String	**参数解释：** 集群节点的变更状态（扩容/缩容/变更规格）。当该参数取值为空时，表示集群节点没有进行变更操作。（默认为英文，当X-Language请求头被设置为zh-cn时，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** ● scaling-out：扩容中。 ● scaling-in：缩容中。 ● scaling-error：处于running状态，且上一次扩容/缩容/升级规格失败的集群。 ● scaling-up：Master节点规格升级中。 ● scaling_up_first：备Master节点规格升级中。 ● scaled_up_first：备Master节点规格升级成功。 ● scaled-up-success：Master节点规格升级成功。 **默认取值：** 不涉及

参数	参数类型	描述
nodeGroups	Array of **NodeGroupV10** objects	**参数解释：** Master节点、Core节点和Task节点列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
taskNodeGroups	Array of **NodeGroupV10** objects	**参数解释：** Task节点列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eipId	String	**参数解释：** 集群弹性公网ip的唯一标识。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eipAddress	String	**参数解释：** 集群弹性公网ip的IPV4地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
eipv6Address	String	**参数解释：** 集群弹性公网ip的IPV6地址，IPv4时无此字段。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-24 ComponentAmb

参数	参数类型	描述
componentId	String	**参数解释：** 组件ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentName	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentVersion	String	**参数解释：** 组件版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
componentDesc	String	**参数解释：** 组件描述信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-25 BootstrapScript

参数	参数类型	描述
name	String	**参数解释：** 引导操作脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的引导操作脚本名称不允许相同。不能以空格开头，只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	参数类型	描述
uri	String	**参数解释：** 引导操作脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。例如输入MRS提供的公共样例脚本路径。示例：s3a://bootstrap/presto/presto-install.sh，其中安装dualroles时，presto-install.sh脚本参数为dualroles,安装worker时，presto-install.sh脚本参数为worker。根据Presto使用习惯，建议您在Active Master节点上安装dualroles，在Core节点上安装worker。 ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	String	**参数解释：** 引导操作脚本参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
nodes	Array of strings	**参数解释：** 引导操作脚本所执行的节点类型，包含master、core和task三种类型。 **约束限制：** 节点类型必须为小写字母。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
active_master	Boolean	**参数解释：** 引导操作脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本只运行在主Master节点上。 ● false：引导操作脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	String	**参数解释：** 引导操作脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此引导操作是否执行成功，则集群都能继续安装和启动。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue
before_component_start	Boolean	**参数解释：** 引导操作脚本执行的时间。目前支持“组件启动前”和“组件启动后”两种类型。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本在组件启动前执行。 ● false：引导操作脚本在组件启动后执行。 **默认取值：** false

参数	参数类型	描述
start_time	Long	**参数解释：** 单个引导操作脚本的执行时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
state	String	**参数解释：** 单个引导操作脚本的运行状态。 **约束限制：** 不涉及 **取值范围：** ● PENDING：挂起 ● IN_PROGRESS：正在处理 ● SUCCESS：处理成功 ● FAILURE：处理失败 **默认取值：** 不涉及
action_stages	Array of strings	**参数解释：** 选择引导操作脚本执行的时间。 ● BEFORE_COMPONENT_FIRST_START: 组件首次启动后 ● AFTER_COMPONENT_FIRST_START: 组件首次启动前 ● BEFORE_SCALE_IN: 缩容前 ● AFTER_SCALE_IN: 缩容后 ● BEFORE_SCALE_OUT: 扩容前 ● AFTER_SCALE_OUT: 扩容后 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-26 NodeGroupV10

参数	参数类型	描述
GroupName	String	**参数解释：** 节点组名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeNum	Integer	**参数解释：** 节点数量，取值范围0-500，Master节点和Core节点数量至少为1，Core与Task节点总数最大为500个。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
NodeSize	String	**参数解释：** 节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeSpecId	String	**参数解释：** 节点实例规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
VmProductId	String	**参数解释：** 节点虚拟机产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
VmSpecCode	String	**参数解释：** 节点虚拟机产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeProductId	String	**参数解释：** 节点实例产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeSize	Integer	**参数解释：** 节点系统盘大小，不可配置，默认为40GB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
RootVolumeProductId	String	**参数解释：** 节点系统盘的产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeType	String	**参数解释：** 节点系统盘的类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeResourceSpecCode	String	**参数解释：** 节点系统盘产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeResourceType	String	**参数解释：** 节点系统盘产品类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
DataVolumeType	String	**参数解释：** 节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO **默认取值：** 不涉及
DataVolumeCount	Integer	**参数解释：** 节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeSize	Integer	**参数解释：** 节点数据磁盘存储大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeProductId	String	**参数解释：** 节点数据磁盘的产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
DataVolumeResourceSpecCode	String	**参数解释：** 节点数据磁盘的产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeResourceType	String	**参数解释：** 节点数据磁盘的产品类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

```
GET /v1.1/{project_id}/cluster_infos?
pageSize={page_size}&currentPage={current_page}&clusterState={cluster_state}&tags={tags}
```

响应示例

状态码： 200

查询集群列表信息成功。

```
{
  "clusterTotal" : "1",
  "clusters" : [ {
    "clusterId" : "bc134369-294c-42b7-a707-b2036ba38524",
    "clusterName" : "mrs_D0zW",
    "masterNodeNum" : "2",
    "coreNodeNum" : "3",
    "clusterState" : "terminated",
    "createAt" : "1498272043",
    "updateAt" : "1498636753",
    "chargingStartTime" : "1498273733",
    "logCollection" : "1",
    "billingType" : "Metered",
    "dataCenter" : "cn-north-1",
    "vpc" : null,
    "duration" : "0",
    "fee" : null,
    "hadoopVersion" : null,
    "masterNodeSize" : null,
    "coreNodeSize" : null,
    "componentList" : [ {
      "componentId" : "MRS 3.1.0_001",
      "componentName" : "Hadoop",
```

```
      "componentVersion" : "3.1.1",
      "componentDesc" : "A framework that allows for the distributed processing of large data sets across
clusters."
    }, {
      "componentId" : "MRS 3.1.0_003",
      "componentName" : "HBase",
      "componentVersion" : "2.1.1",
      "componentDesc" : "A scalable, distributed database that supports structured data storage for large
tables."
    }, {
      "componentId" : "MRS 3.1.0_002",
      "componentName" : "Spark",
      "componentVersion" : "2.3.2",
      "componentDesc" : "A fast and general engine for large-scale data processing."
    }, {
      "componentId" : "MRS 3.1.0_004",
      "componentName" : "Hive",
      "componentVersion" : "3.1.0",
      "componentDesc" : "A data warehouse infrastructure that provides data summarization and ad hoc
querying."
    } ],
    "externalIp" : null,
    "externalAlternateIp" : null,
    "internalIp" : null,
    "deploymentId" : null,
    "remark" : "",
    "orderId" : null,
    "azId" : null,
    "azCode" : null,
    "masterNodeProductId" : null,
    "masterNodeSpecId" : null,
    "coreNodeProductId" : null,
    "coreNodeSpecId" : null,
    "azName" : "az1.dc1",
    "instanceId" : null,
    "vnc" : "v2/5a3314075bfa49b9ae360f4ecd333695/servers/e2cda891-232e-4703-995e-3b1406add01d/
action",
    "tenantId" : null,
    "volumeSize" : "0",
    "volumeType" : null,
    "subnetId" : null,
    "subnetName" : null,
    "securityGroupsId" : null,
    "slaveSecurityGroupsId" : null,
    "bootstrapScripts" : [ {
      "name" : "test1-success",
      "uri" : "s3a://bootscript/script/simple/basic_success.sh",
      "parameters" : "",
      "nodes" : [ "master", "core" ],
      "active_master" : true,
      "fail_action" : "errorout",
      "before_component_start" : true,
      "state" : "SUCCESS",
      "start_time" : 1527681083,
      "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
    } ],
    "isMrsManagerFinish" : false,
    "stageDesc" : "Installing MRS Manager",
    "safeMode" : "0",
    "clusterVersion" : null,
    "nodePublicCertName" : null,
    "masterNodeIp" : "unknown",
    "privateIpFirst" : null,
    "eipId" : "b16dd5eb-5e5b-486a-906a-2e8f6e814a7a",
    "eipAddress" : "100.XXX.XXX.XXX",
    "eipv6Address" : "2403:XXXX:XXXX::XXXX:XXXX",
    "errorInfo" : "",
    "clusterType" : "0",
    "nodeGroups" : [ {
```



```
  "GroupName" : "master_node_default_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
 }, {
  "GroupName" : "core_node_analysis_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
 }, {
  "GroupName" : "task_node_analysis_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
 } ],
 "taskNodeGroups" : [ {
  "GroupName" : "task_node_default_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
```



```
    "RootVolumeResourceType" : "",
    "DataVolumeType" : "SATA",
    "DataVolumeCount" : "1",
    "DataVolumeSize" : "100",
    "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
    "DataVolumeResourceSpecCode" : "",
    "DataVolumeResourceType" : ""
   } ],
   "masterDataVolumeType" : "SATA",
   "masterDataVolumeSize" : "200",
   "masterDataVolumeCount" : "1",
   "coreDataVolumeType" : "SATA",
   "coreDataVolumeSize" : "100",
   "coreDataVolumeCount" : "1",
   "periodType" : "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.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .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();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
```



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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

    client = MrsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(MrsRegion.value_of("<YOUR REGION>")) \
        .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"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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



```
        WithProjectId(projectId).
        Build()

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            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	查询集群列表信息成功。

错误码

请参见**错误码**。

7.1.4 查询集群详情

功能介绍

查看指定集群的详细信息。该接口不兼容Sahara。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/cluster_infos/{cluster_id}

表 7-27 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 7-28 响应 Body 参数

参数	参数类型	描述
cluster	**Cluster** object	**参数解释：** 集群参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-29 Cluster

参数	参数类型	描述
clusterId	String	**参数解释：** 集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
clusterName	String	**参数解释：** 集群名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
totalNodeNum	String	**参数解释：** 集群部署的节点总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
clusterState	String	**参数解释：** 集群状态（默认为英文，当X-Language请求头被设置为zh-cn，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** ● starting：启动中。 ● running：运行中。 ● terminated：已删除。 ● failed：失败。 ● abnormal：异常。 ● terminating：删除中。 ● frozen：已冻结。 ● scaling-out：扩容中。 ● scaling-in：缩容中。 **默认取值：** 不涉及

参数	参数类型	描述
stageDesc	String	**参数解释：** 集群进度描述。（默认为英文，当X-Language请求头被设置为zh-cn时，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** 安装集群进度包括： ● Verifying cluster parameters：校验集群参数中 ● Applying for cluster resources：申请集群资源中 ● Creating VM：创建虚拟机中 ● Initializing VM：初始化虚拟机中 ● Installing MRS Manager：安装MRS Manager中 ● Deploying cluster：部署集群中 ● Cluster installation failed：集群安装失败 扩容集群进度包括： ● Preparing for cluster expansion：准备扩容中 ● Creating VM：创建虚拟机中 ● Initializing VM：初始化虚拟机中 ● Adding node to the cluster：节点加入集群中 ● Cluster expansion failed：集群扩容失败 缩容集群进度包括： ● Preparing for cluster shrink：正在准备缩容 ● Decommissioning instance：实例退服中 ● Deleting VM：删除虚拟机中 ● Deleting node from the cluster：从集群删除节点中 ● Cluster shrink failed：集群缩容失败 集群安装、扩容、缩容失败，stageDesc会显示失败的原因。 **默认取值：** 不涉及

参数	参数类型	描述
createAt	String	**参数解释：** 集群创建时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
updateAt	String	**参数解释：** 集群更新时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
chargingStartTime	String	**参数解释：** 开始计费时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
billingType	String	**参数解释：** 集群计费模式。 **约束限制：** 不涉及 **取值范围：** ● 包周期计费 ● 按需计费 **默认取值：** 不涉及

参数	参数类型	描述
dataCenter	String	**参数解释：** 集群工作区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpc	String	**参数解释：** VPC名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vpcId	String	**参数解释：** VPC ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
duration	String	**参数解释：** 集群购买时长。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
fee	String	**参数解释：** 创建集群所需费用，系统自动计算。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
hadoopVersion	String	**参数解释：** Hadoop组件版本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentList	Array of **ComponentAmb** objects	**参数解释：** 组件列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
externalIp	String	**参数解释：** 公网IP地址。用于集群外访问集群主master进程所开放IP的地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
externalAlternateIp	String	**参数解释：** 公网备用IP地址。用于集群外访问集群备master进程所使用的IP地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
internalIp	String	**参数解释：** 内网IP地址。用于集群内访问集群master进程所使用IP的地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
deploymentId	String	**参数解释：** 集群部署ID。本集群的任务部署所对应的ID，用于云服务查询集群部署任务详情。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
remark	String	**参数解释：** 集群备注信息。当前默认都为空。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
orderId	String	**参数解释：** 创建集群的订单号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azId	String	**参数解释：** 可用区域ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azName	String	**参数解释：** 可用区域名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
azCode	String	**参数解释：** 可用区域英文名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
availabilityZoneId	String	**参数解释：** 可用区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
instanceId	String	**参数解释：** 实例ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
vnc	String	**参数解释：** 远程登录弹性云服务器的URI地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenantId	String	**参数解释：** 项目编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
volumeSize	Integer	**参数解释：** 磁盘存储空间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
volumeType	String	**参数解释：** 磁盘类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
subnetId	String	**参数解释：** 子网ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
subnetName	String	**参数解释：** 子网名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
securityGroupsId	String	**参数解释：** 安全组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
slaveSecurityGroupsId	String	**参数解释：** 非Master节点的安全组id，当前一个MRS集群只会使用一个安全组，所以该字段已经废弃，从兼容性考虑，该字段会返回和securityGroupsId同样的值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
bootstrapScripts	Array of **BootstrapScript** objects	**参数解释：** 配置引导操作脚本信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
safeMode	Integer	**参数解释：** MRS集群运行模式。 **约束限制：** 不涉及 **取值范围：** ● 0：普通集群 ● 1：安全集群 **默认取值：** 不涉及

参数	参数类型	描述
clusterVersion	String	**参数解释：** 集群版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
nodePublicCertName	String	**参数解释：** 密钥文件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeIp	String	**参数解释：** Master节点IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
privateIpFirst	String	**参数解释：** 首选私有IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
errorInfo	String	**参数解释：** 错误信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	String	**参数解释：** 标签信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeNum	String	**参数解释：** 集群部署的Master节点数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeNum	String	**参数解释：** 集群部署的Core节点数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
masterNodeSize	String	**参数解释：** Master节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeSize	String	**参数解释：** Core节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeProductId	String	**参数解释：** Master节点产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterNodeSpecId	String	**参数解释：** Master节点规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
coreNodeProductId	String	**参数解释：** Core节点产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreNodeSpecId	String	**参数解释：** Core节点规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterDataVolumeType	String	**参数解释：** Master节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
masterDataVolumeSize	Integer	**参数解释：** Master节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。单位为GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及

参数	参数类型	描述
masterDataVolumeCount	Integer	**参数解释：** Master节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 只能为1。 **默认取值：** 不涉及
coreDataVolumeType	String	**参数解释：** Core节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
coreDataVolumeSize	Integer	**参数解释：** Core节点数据磁盘存储空间。为增大数据存储容量，创建集群时可同时添加磁盘。单位为GB。 **约束限制：** 不涉及 **取值范围：** 100-32000 **默认取值：** 不涉及
coreDataVolumeCount	Integer	**参数解释：** Core节点数据磁盘个数。 **约束限制：** 不涉及 **取值范围：** 1-10 **默认取值：** 不涉及

参数	参数类型	描述
enterpriseProjectId	String	**参数解释：** 企业项目ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
isMrsManagerFinish	Boolean	**参数解释：** 表示集群创建过程中，MRS Manager是否安装完成。 **约束限制：** 不涉及 **取值范围：** ● true：安装完成。 ● false：安装未完成。 **默认取值：** 不涉及
clusterType	Integer	**参数解释：** 集群类型。 **约束限制：** 不涉及 **取值范围：** ● 0：分析集群。 ● 1：流式集群。 ● 2：混合集群。 ● 3：自定义集群。 ● 4：离线集群。 **默认取值：** 不涉及

参数	参数类型	描述
logCollection	Integer	**参数解释：** 集群安装失败时，是否搜集日志。 **约束限制：** 不涉及 **取值范围：** ● 0：不收集。 ● 1：收集。 **默认取值：** 不涉及
periodType	Integer	**参数解释：** 区分包周期，集群是包年还是包月。 **约束限制：** 不涉及 **取值范围：** ● 0：包月。 ● 1：包年。 **默认取值：** 不涉及
scale	String	**参数解释：** 集群节点的变更状态（扩容/缩容/变更规格）。当该参数取值为空时，表示集群节点没有进行变更操作。（默认为英文，当X-Language请求头被设置为zh-cn时，该字段会返回中文，因此请谨慎设置X-Language等语言请求头）。 **约束限制：** 不涉及 **取值范围：** ● scaling-out：扩容中。 ● scaling-in：缩容中。 ● scaling-error：处于running状态，且上一次扩容/缩容/升级规格失败的集群。 ● scaling-up：Master节点规格升级中。 ● scaling_up_first：备Master节点规格升级中。 ● scaled_up_first：备Master节点规格升级成功。 ● scaled-up-success：Master节点规格升级成功。 **默认取值：** 不涉及

参数	参数类型	描述
nodeGroups	Array of **NodeGroupV10** objects	**参数解释：** Master节点、Core节点和Task节点列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
taskNodeGroups	Array of **NodeGroupV10** objects	**参数解释：** Task节点列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eipId	String	**参数解释：** 集群弹性公网ip的唯一标识。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
eipAddress	String	**参数解释：** 集群弹性公网ip的IPV4地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
eipv6Address	String	**参数解释：** 集群弹性公网ip的IPV6地址，IPv4时无此字段。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-30 ComponentAmb

参数	参数类型	描述
componentId	String	**参数解释：** 组件ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentName	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
componentVersion	String	**参数解释：** 组件版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
componentDesc	String	**参数解释：** 组件描述信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-31 BootstrapScript

参数	参数类型	描述
name	String	**参数解释：** 引导操作脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的引导操作脚本名称不允许相同。不能以空格开头，只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	参数类型	描述
uri	String	**参数解释：** 引导操作脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。例如输入MRS提供的公共样例脚本路径。示例：s3a://bootstrap/presto/presto-install.sh，其中安装dualroles时，presto-install.sh脚本参数为dualroles,安装worker时，presto-install.sh脚本参数为worker。根据Presto使用习惯，建议您在Active Master节点上安装dualroles，在Core节点上安装worker。 ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	String	**参数解释：** 引导操作脚本参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
nodes	Array of strings	**参数解释：** 引导操作脚本所执行的节点类型，包含master、core和task三种类型。 **约束限制：** 节点类型必须为小写字母。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
active_master	Boolean	**参数解释：** 引导操作脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本只运行在主Master节点上。 ● false：引导操作脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	String	**参数解释：** 引导操作脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此引导操作是否执行成功，则集群都能继续安装和启动。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue
before_component_start	Boolean	**参数解释：** 引导操作脚本执行的时间。目前支持“组件启动前”和“组件启动后”两种类型。 **约束限制：** 不涉及 **取值范围：** ● true：引导操作脚本在组件启动前执行。 ● false：引导操作脚本在组件启动后执行。 **默认取值：** false

参数	参数类型	描述
start_time	Long	**参数解释：** 单个引导操作脚本的执行时间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
state	String	**参数解释：** 单个引导操作脚本的运行状态。 **约束限制：** 不涉及 **取值范围：** ● PENDING：挂起 ● IN_PROGRESS：正在处理 ● SUCCESS：处理成功 ● FAILURE：处理失败 **默认取值：** 不涉及
action_stages	Array of strings	**参数解释：** 选择引导操作脚本执行的时间。 ● BEFORE_COMPONENT_FIRST_START: 组件首次启动后 ● AFTER_COMPONENT_FIRST_START: 组件首次启动前 ● BEFORE_SCALE_IN: 缩容前 ● AFTER_SCALE_IN: 缩容后 ● BEFORE_SCALE_OUT: 扩容前 ● AFTER_SCALE_OUT: 扩容后 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-32 NodeGroupV10

参数	参数类型	描述
GroupName	String	**参数解释：** 节点组名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeNum	Integer	**参数解释：** 节点数量，取值范围0-500，Master节点和Core节点数量至少为1，Core与Task节点总数最大为500个。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
NodeSize	String	**参数解释：** 节点的实例规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeSpecId	String	**参数解释：** 节点实例规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
VmProductId	String	**参数解释：** 节点虚拟机产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
VmSpecCode	String	**参数解释：** 节点虚拟机产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
NodeProductId	String	**参数解释：** 节点实例产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeSize	Integer	**参数解释：** 节点系统盘大小，不可配置，默认为40GB。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
RootVolumeProductId	String	**参数解释：** 节点系统盘的产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeType	String	**参数解释：** 节点系统盘的类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeResourceSpecCode	String	**参数解释：** 节点系统盘产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
RootVolumeResourceType	String	**参数解释：** 节点系统盘产品类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
DataVolumeType	String	**参数解释：** 节点数据磁盘存储类别，目前支持SATA、SAS和SSD。 **约束限制：** 不涉及 **取值范围：** ● SATA：普通IO ● SAS：高IO ● SSD：超高IO **默认取值：** 不涉及
DataVolumeCount	Integer	**参数解释：** 节点数据磁盘存储数目。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeSize	Integer	**参数解释：** 节点数据磁盘存储大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeProductId	String	**参数解释：** 节点数据磁盘的产品ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
DataVolumeResourceSpecCode	String	**参数解释：** 节点数据磁盘的产品规格。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
DataVolumeResourceType	String	**参数解释：** 节点数据磁盘的产品类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

```
GET /v1.1/{project_id}/cluster_infos/{cluster_id}
```

响应示例

状态码： 200

查询集群详情成功。

```
{
  "cluster" : {
    "clusterId" : "bdb064ff-2855-4624-90d5-e9a6376abd6e",
    "clusterName" : "c17022001",
    "masterNodeNum" : "2",
    "coreNodeNum" : "3",
    "clusterState" : "scaling-in",
    "stageDesc" : "Installing MRS Manager",
    "createAt" : "1487570757",
    "updateAt" : "1487668974",
    "billingType" : "Metered",
    "dataCenter" : "cn-north-1",
    "vpc" : "vpc-autotest",
    "vpcId" : "e2978efd-ca12-4058-9332-1ca0bfbab592",
    "duration" : "0",
    "fee" : "0",
    "hadoopVersion" : "",
    "masterNodeSize" : "s3.2xlarge.2.linux.bigdata",
    "coreNodeSize" : "s1.xlarge.linux.bigdata",
    "componentList" : [ {
      "componentId" : "MRS 3.1.0_001",
      "componentName" : "Hadoop",
      "componentVersion" : "3.1.1",
      "componentDesc" : "A framework that allows for the distributed processing of large data sets across
```

```
clusters."
    }, {
      "componentId" : "MRS 3.1.0_002",
      "componentName" : "Spark",
      "componentVersion" : "2.3.2",
      "componentDesc" : "A fast and general engine for large-scale data processing."
    }, {
      "componentId" : "MRS 3.1.0_004",
      "componentName" : "Hive",
      "componentVersion" : "3.1.0",
      "componentDesc" : "A data warehouse infrastructure that provides data summarization and ad hoc
querying."
    }, {
      "componentId" : "MRS 3.1.0_003",
      "componentName" : "HBase",
      "componentVersion" : "2.1.1",
      "componentDesc" : "A scalable, distributed database that supports structured data storage for large
tables."
    } ],
    "externalIp" : "100.XXX.XXX.XXX",
    "externalAlternateIp" : "100.XXX.XXX.XXX",
    "internalIp" : "192.XXX.XXX.XXX",
    "eipId" : "b16dd5eb-5e5b-486a-906a-2e8f6e814a7a",
    "eipAddress" : "100.XXX.XXX.XXX",
    "eipv6Address" : "2403:XXXX:XXXX::XXXX:XXXX",
    "deploymentId" : "4ac46ca7-a488-4b91-82c2-e4d7aa9c40c2",
    "remark" : "",
    "orderId" : "null",
    "azId" : "null",
    "masterNodeProductId" : "b35cf2d2348a445ca74b32289a160882",
    "masterNodeSpecId" : "8ab05e503b4c42abb304e2489560063b",
    "coreNodeProductId" : "dc970349d128460e960a0c2b826c427c",
    "coreNodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
    "azName" : "az1.dc1",
    "instanceId" : "4ac46ca7-a488-4b91-82c2-e4d7aa9c40c2",
    "vnc" : null,
    "tenantId" : "3f99e3319a8943ceb15c584f3325d064",
    "volumeSize" : "100",
    "volumeType" : "SATA",
    "subnetId" : "6b96eec3-4f8d-4c83-93e2-6ec625001d7c",
    "subnetName" : "subnet-ftest",
    "securityGroupsId" : "930e34e2-195d-401f-af07-0b64ea6603f8",
    "slaveSecurityGroupsId" : "2ef3343e-3477-4a0d-80fe-4d874e4f81b8",
    "bootstrapScripts" : [ {
      "name" : "test1-success",
      "uri" : "s3a://bootscript/script/simple/basic_success.sh",
      "parameters" : "",
      "nodes" : [ "master", "core" ],
      "active_master" : true,
      "fail_action" : "errorout",
      "before_component_start" : true,
      "state" : "SUCCESS",
      "start_time" : 1527681083,
      "action_stages" : [ "AFTER_SCALE_IN", "AFTER_SCALE_OUT" ]
    } ],
    "isMrsManagerFinish" : false,
    "safeMode" : "1",
    "clusterVersion" : "MRS 2.1.0",
    "nodePublicCertName" : "myp",
    "masterNodeIp" : "192.XXX.XXX.XXX",
    "privateIpFirst" : "192.XXX.XXX.XXX",
    "errorInfo" : null,
    "tags" : "k1=v1,k2=v2,k3=v3",
    "clusterType" : "",
    "logCollection" : "1",
    "nodeGroups" : [ {
      "GroupName" : "master_node_default_group",
      "NodeNum" : "1",
      "NodeSize" : "s3.xlarge.2.linux.bigdata",
```



```
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
}, {
  "GroupName" : "core_node_analysis_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
}, {
  "GroupName" : "task_node_analysis_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
  "DataVolumeSize" : "100",
  "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "DataVolumeResourceSpecCode" : "",
  "DataVolumeResourceType" : ""
} ],
"taskNodeGroups" : [ {
  "GroupName" : "task_node_default_group",
  "NodeNum" : "1",
  "NodeSize" : "s3.xlarge.2.linux.bigdata",
  "NodeSpecId" : "cdc6035a249a40249312f5ef72a23cd7",
  "VmProductId" : "",
  "VmSpecCode" : null,
  "NodeProductId" : "dc970349d128460e960a0c2b826c427c",
  "RootVolumeSize" : "40",
  "RootVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
  "RootVolumeType" : "SATA",
  "RootVolumeResourceSpecCode" : "",
  "RootVolumeResourceType" : "",
  "DataVolumeType" : "SATA",
  "DataVolumeCount" : "1",
```



```
    "DataVolumeSize" : "100",
    "DataVolumeProductId" : "16c1dcf0897249758b1ec276d06e0572",
    "DataVolumeResourceSpecCode" : "",
    "DataVolumeResourceType" : ""
   } ],
   "masterDataVolumeType" : "SATA",
   "masterDataVolumeSize" : "200",
   "masterDataVolumeCount" : "1",
   "coreDataVolumeType" : "SATA",
   "coreDataVolumeSize" : "100",
   "coreDataVolumeCount" : "1",
   "periodType" : "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.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ShowClusterDetailsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowClusterDetailsRequest request = new ShowClusterDetailsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ShowClusterDetailsResponse response = client.showClusterDetails(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
```



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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowClusterDetailsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.show_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"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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



```
        Build()

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，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。

错误码

请参见**错误码**。

7.1.5 查询主机列表

功能介绍

该接口用于查询输入集群的主机列表详情。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/clusters/{cluster_id}/hosts

表 7-33 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

表 7-34 Query 参数

参数	是否必选	参数类型	描述
pageSize	否	String	**参数解释：** 分页查询每页返回的最大集群数量。 **约束限制：** 不涉及 **取值范围：** 1-2147483646 **默认取值：** 10

参数	是否必选	参数类型	描述
currentPage	否	String	**参数解释：** 当前查询页码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 1

请求参数

无

响应参数

状态码： 200

表 7-35 响应 Body 参数

参数	参数类型	描述
hosts	Array of **HostModel** objects	**参数解释：** 主机参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
total	Integer	**参数解释：** 主机列表总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-36 HostModel

参数	参数类型	描述
id	String	**参数解释：** 虚拟机ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
name	String	**参数解释：** 虚拟机名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
ip	String	**参数解释：** 虚拟机IP地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
availability_zone_id	String	**参数解释：** 可用区域。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
tags	Array of **TagPlain** objects	**参数解释：** 标签列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	String	**参数解释：** 虚拟机当前状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_id	String	**参数解释：** 节点资源ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
flavor	String	**参数解释：** 虚拟机规格ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
type	String	**参数解释：** 虚拟机类型，当前支持MasterNode，CoreNode，TaskNode。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
mem	String	**参数解释：** 内存。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cpu	String	**参数解释：** CPU核数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
root_volume_size	String	**参数解释：** 操作系统盘容量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
data_volume_type	String	**参数解释：** 数据盘类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
data_volume_size	Integer	**参数解释：** 数据盘容量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
data_volume_count	Integer	**参数解释：** 数据盘个数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_group_name	String	**参数解释：** 节点组名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-37 TagPlain

参数	参数类型	描述
key	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

请求示例

```
GET /v1.1/{project_id}/clusters/{cluster_id}/hosts
```

响应示例

状态码： 200

查询主机列表信息成功。

```
{
  "total" : "5",
  "hosts" : [ {
    "id" : "063d1d47-ae91-4a48-840c-b3cfe4efbcf0",
    "name" : "a78e161c-d14f-4b68-8c2d-0219920ce844_node_core_IQhiC",
    "ip" : "192.168.0.169",
    "availability_zone_id" : null,
    "tags" : null,
    "status" : "ACTIVE",
    "resource_id" : "95c1eabc-ed1d-4037-97d1-62f0587790c7",
    "flavor" : "c2.2xlarge.linux.mrs",
    "type" : "Core",
    "mem" : "16384",
    "cpu" : "8",
    "root_volume_size" : "480",
    "data_volume_type" : "SATA",
    "data_volume_size" : "600",
```

```
  "data_volume_count" : "1",
  "node_group_name" : "node_group_1"
}, {
  "id" : "dc5c6208-faa2-4727-a65a-2b1ce235d350",
  "name" : "a78e161c-d14f-4b68-8c2d-0219920ce844_node_master1_ASzkl",
  "ip" : "192.168.0.156",
  "availability_zone_id" : null,
  "tags" : null,
  "status" : "ACTIVE",
  "resource_id" : "95c1eabc-ed1d-4037-97d1-62f0587790c7",
  "flavor" : "c2.4xlarge.linux.mrs",
  "type" : "Master",
  "mem" : "32768",
  "cpu" : "16",
  "root_volume_size" : "480",
  "data_volume_type" : "SATA",
  "data_volume_size" : "600",
  "data_volume_count" : "1",
  "node_group_name" : "master_node_default_group"
}, {
  "id" : "c0ce793d-848b-448a-835b-ea0cac534b09",
  "name" : "a78e161c-d14f-4b68-8c2d-0219920ce844_node_core_ANnRN",
  "ip" : "192.168.0.243",
  "availability_zone_id" : null,
  "tags" : null,
  "status" : "ACTIVE",
  "resource_id" : "95c1eabc-ed1d-4037-97d1-62f0587790c7",
  "flavor" : "c2.2xlarge.linux.mrs",
  "type" : "Core",
  "mem" : "16384",
  "cpu" : "8",
  "root_volume_size" : "480",
  "data_volume_type" : "SATA",
  "data_volume_size" : "600",
  "data_volume_count" : "1",
  "node_group_name" : "node_group_1"
}, {
  "id" : "95c23e43-ef6e-4732-b6ed-a5f1c7779fae",
  "name" : "a78e161c-d14f-4b68-8c2d-0219920ce844_node_core_uRRiA",
  "ip" : "192.168.0.126",
  "availability_zone_id" : null,
  "tags" : null,
  "status" : "ACTIVE",
  "resource_id" : "95c1eabc-ed1d-4037-97d1-62f0587790c7",
  "flavor" : "c2.2xlarge.linux.mrs",
  "type" : "Core",
  "mem" : "16384",
  "cpu" : "8",
  "root_volume_size" : "480",
  "data_volume_type" : "SATA",
  "data_volume_size" : "600",
  "data_volume_count" : "1",
  "node_group_name" : "node_group_1"
}, {
  "id" : "63bdbf75-1133-4a94-8c27-1fa12c8b9e70",
  "name" : "a78e161c-d14f-4b68-8c2d-0219920ce844_node_master2_StqFu",
  "ip" : "192.168.0.22",
  "availability_zone_id" : null,
  "tags" : null,
  "status" : "ACTIVE",
  "resource_id" : "95c1eabc-ed1d-4037-97d1-62f0587790c7",
  "flavor" : "c2.4xlarge.linux.mrs",
  "type" : "Master",
  "mem" : "32768",
  "cpu" : "16",
  "root_volume_size" : "480",
  "data_volume_type" : "SATA",
  "data_volume_size" : "600",
  "data_volume_count" : "1",
```



```
    "node_group_name" : "master_node_default_group"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ListHostsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListHostsRequest request = new ListHostsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListHostsResponse response = client.listHosts(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
```



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListHostsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_hosts(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListHostsRequest{}
    request.ClusterId = "{cluster_id}"
```



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

更多

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

状态码

状态码	描述
200	查询主机列表信息成功。

错误码

请参见**错误码**。

7.1.6 删除集群

功能介绍

数据完成处理分析后或者集群运行异常无法提供服务时可删除集群服务。该接口兼容Sahara。

处于如下状态的集群不允许删除：

- scaling-out：扩容中
- scaling-in：缩容中
- starting：启动中
- terminating：删除中
- terminated：已删除
- failed：失败

接口约束

无

调用方法

请参见**如何调用API**。

URI

DELETE /v1.1/{project_id}/clusters/{cluster_id}

表 7-38 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

无

请求示例

```
DELETE /v1.1/{project_id}/clusters/{cluster_id}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.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");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteClusterRequest request = new DeleteClusterRequest();
        request.withClusterId("{cluster_id}");
        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

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteClusterRequest()
        request.cluster_id = "{cluster_id}"
        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"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteClusterRequest{}
    request.ClusterId = "{cluster_id}"
    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
204	删除集群成功。

错误码

请参见**错误码**。

7.2 弹性伸缩接口

7.2.1 配置弹性伸缩规则

功能介绍

对弹性伸缩规则进行编辑。

在创建集群并执行作业接口中也可以创建弹性伸缩规则。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/autoscaling-policy/{cluster_id}

表 7-39 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-40 请求 Body 参数

参数	是否必选	参数类型	描述
node_group	是	String	**参数解释：** 弹性伸缩规则适用的节点类型，当前只支持task节点。 **约束限制：** 不涉及 **取值范围：** • task_node_default_group：弹性伸缩规则适用task节点。 **默认取值：** 不涉及
auto_scaling_policy	是	**AutoScalingPolicy** object	**参数解释：** 弹性伸缩规则。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-41 AutoScalingPolicy

参数	是否必选	参数类型	描述
auto_scaling_enable	是	Boolean	**参数解释：** 当前自动伸缩规则是否开启。 **约束限制：** 不涉及 **取值范围：** • true：开启自动伸缩规则。 • false：不开启自动伸缩规则。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
min_capacity	是	Integer	**参数解释：** 指定该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 指定该节点组的最大节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
resources_plans	否	Array of **ResourcesPlan** objects	**参数解释：** 资源计划列表。若该参数为空表示不启用资源计划。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过5条。 **取值范围：** 不涉及 **默认取值：** 不涉及
rules	否	Array of **Rule** objects	**参数解释：** 自动伸缩的规则列表。 **约束限制：** 当启用弹性伸缩时，资源计划与自动伸缩规则需至少配置其中一种。不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
exec_scripts	否	Array of ScaleScript objects	**参数解释：** 弹性伸缩自定义自动化脚本列表。若该参数为空表示不启用自动化脚本。 **约束限制：** 不能超过10条。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-42 ResourcesPlan

参数	是否必选	参数类型	描述
period_type	是	String	**参数解释：** 资源计划的周期类型，当前只允许以下类型：daily。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
start_time	是	String	**参数解释：** 资源计划的起始时间。格式为“hour:minute”，表示时间在0:00-23:59之间。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
end_time	是	String	**参数解释：** 资源计划的结束时间，格式与“start_time”相同。 **约束限制：** 不早于start_time表示的时间，且与start_time间隔不小于30min。 **取值范围：** 不涉及 **默认取值：** 不涉及
min_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最小保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及
max_capacity	是	Integer	**参数解释：** 资源计划内该节点组的最大保留节点数。 **约束限制：** 不涉及 **取值范围：** 0-500 **默认取值：** 不涉及

表 7-43 Rule

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩规则的名称。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。在一个节点组范围内，不允许重名。 **默认取值：** 不涉及
description	否	String	**参数解释：** 弹性伸缩规则的说明。 **约束限制：** 不涉及 **取值范围：** 长度小于等于1024个字符。 **默认取值：** 不涉及
adjustment_type	是	String	**参数解释：** 弹性伸缩规则的调整类型。 **约束限制：** 不涉及 **取值范围：** ● scale_out：扩容 ● scale_in：缩容 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
cool_down_minutes	是	Integer	**参数解释：** 触发弹性伸缩规则后，该集群处于冷却状态（不再执行弹性伸缩操作）的时长，单位为分钟。 **约束限制：** 不涉及 **取值范围：** 0-10080。10080为一周的分钟数。 **默认取值：** 不涉及
scaling_adjustment	是	Integer	**参数解释：** 单次调整集群节点的个数。 **约束限制：** 不涉及 **取值范围：** 1-100 **默认取值：** 不涉及
trigger	是	**Trigger** object	**参数解释：** 描述该规则触发条件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-44 Trigger

参数	是否必选	参数类型	描述
metric_name	是	String	**参数解释：** 指标名称。该触发条件会依据该名称对应指标的值来进行判断。 **约束限制：** 不涉及 **取值范围：** 取值范围请参见**"弹性伸缩指标列表"**。 **默认取值：** 不涉及
metric_value	是	String	**参数解释：** 指标阈值。触发该条件的指标阈值，只允许输入整数或者带两位小数的数。 **约束限制：** 不涉及 **取值范围：** 只允许输入整数或者带两位小数的数。 **默认取值：** 不涉及
comparison_operator	否	String	**参数解释：** 指标判断逻辑运算符。 **约束限制：** 不涉及 **取值范围：** ● LT：小于 ● GT：大于 ● LTOE：小于等于 ● GTOE：大于等于 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
evaluation_periods	是	Integer	**参数解释：** 判断连续满足指标阈值的周期数(一个周期为5分钟)。 **约束限制：** 不涉及 **取值范围：** 1-200 **默认取值：** 不涉及

表 7-45 ScaleScript

参数	是否必选	参数类型	描述
name	是	String	**参数解释：** 弹性伸缩自定义自动化脚本的名称。 **约束限制：** 不涉及 **取值范围：** 同一个集群的自定义自动化脚本名称不允许相同。只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
uri	是	String	**参数解释：** 自定义自动化脚本的路径。设置为OBS桶的路径或虚拟机本地的路径。 ● OBS桶的路径：直接手动输入脚本路径。示例：s3a://XXX/scale.sh ● 虚拟机本地的路径：用户需要输入正确的脚本路径。脚本所在的路径必须以‘/’开头，以.sh结尾。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
parameters	否	String	**参数解释：** 自定义自动化脚本参数。多个参数间用空格隔开。可以传入以下系统预定义参数： ● ${mrs_scale_node_num}：扩缩容节点数 ● ${mrs_scale_type}：扩缩容类型，扩容为scale_out，缩容为scale_in ● ${mrs_scale_node_hostnames}：扩缩容的节点主机名称 ● ${mrs_scale_node_ips}：扩缩容的节点IP ● ${mrs_scale_rule_name}：触发扩缩容的规则名 其他用户自定义参数使用方式与普通shell脚本相同，多个参数中间用空格隔开。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
nodes	是	Array of strings	**参数解释：** 自定义自动化脚本所执行的节点组名称（非自定义集群也可使用节点类型，包含Master、Core和Task三种类型）。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
active_master	否	Boolean	**参数解释：** 自定义自动化脚本是否只运行在主Master节点上。 **约束限制：** 不涉及 **取值范围：** ● true：自定义自动化脚本只运行在主Master节点上。 ● false：自定义自动化脚本可运行在所有Master节点上。 **默认取值：** false
fail_action	是	String	**参数解释：** 自定义自动化脚本执行失败后，是否继续执行后续脚本和创建集群。建议您在调试阶段设置为“continue”，无论此自定义自动化脚本是否执行成功，则集群都能继续安装和启动。由于缩容成功无法回滚，因此缩容后执行的脚本“fail_action”必须设置为“continue”。 **约束限制：** 不涉及 **取值范围：** ● continue：继续执行后续脚本。 ● errorout：终止操作。 **默认取值：** continue

参数	是否必选	参数类型	描述
action_stage	是	String	**参数解释：** 脚本执行时机。 **约束限制：** 不涉及 **取值范围：** • before_scale_out：扩容前 • before_scale_in：缩容前 • after_scale_out：扩容后 • after_scale_in：缩容后 **默认取值：** 不涉及

响应参数

状态码： 200

表 7-46 响应 Body 参数

参数	参数类型	描述
result	String	**参数解释：** 操作结果。 **约束限制：** • succeeded：操作成功。 • 操作失败时返回的错误码信息如**错误码**所示。 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

配置集群弹性伸缩规则

```
POST https://{endpoint}/v1.1/{project_id}/autoscaling-policy/{cluster_id}

{
  "node_group" : "task_node_analysis_group",
  "auto_scaling_policy" : {
    "auto_scaling_enable" : "true",
    "min_capacity" : "1",
    "max_capacity" : "3",
    "resources_plans" : [ {
      "period_type" : "daily",
```

```
    "start_time" : "9:50",
    "end_time" : "10:20",
    "min_capacity" : "2",
    "max_capacity" : "3"
   }, {
    "period_type" : "daily",
    "start_time" : "10:20",
    "end_time" : "12:30",
    "min_capacity" : "0",
    "max_capacity" : "2"
   } ],
   "exec_scripts" : [ {
    "name" : "before_scale_out",
    "uri" : "s3a://XXX/zeppelin_install.sh",
    "parameters" : "${mrs_scale_node_num} ${mrs_scale_type} xxx",
    "nodes" : [ "master_node_default_group", "core_node_analysis_group", "task_node_analysis_group" ],
    "active_master" : "true",
    "action_stage" : "before_scale_out",
    "fail_action" : "continue"
   }, {
    "name" : "after_scale_out",
    "uri" : "s3a://XXX/storm_rebalance.sh",
    "parameters" : "${mrs_scale_node_hostnames} ${mrs_scale_node_ips}",
    "nodes" : [ "master_node_default_group", "core_node_analysis_group", "task_node_analysis_group" ],
    "active_master" : "true",
    "action_stage" : "after_scale_out",
    "fail_action" : "continue"
   } ],
   "rules" : [ {
    "name" : "default-expand-1",
    "adjustment_type" : "scale_out",
    "cool_down_minutes" : "5",
    "scaling_adjustment" : "1",
    "trigger" : {
     "metric_name" : "YARNMemoryAvailablePercentage",
     "metric_value" : "25",
     "comparison_operator" : "LT",
     "evaluation_periods" : "10"
    }
   }, {
    "name" : "default-shrink-1",
    "adjustment_type" : "scale_in",
    "cool_down_minutes" : "5",
    "scaling_adjustment" : "1",
    "trigger" : {
     "metric_name" : "YARNMemoryAvailablePercentage",
     "metric_value" : "70",
     "comparison_operator" : "GT",
     "evaluation_periods" : "10"
    }
   } ]
  }
}
```

响应示例

状态码： 200

操作成功。

```
{
  "result" : "succeeded"
}
```

SDK 代码示例

SDK代码示例如下。

Java

配置集群弹性伸缩规则

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

public class CreateScalingPolicySolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateScalingPolicyRequest request = new CreateScalingPolicyRequest();
        request.withClusterId("{cluster_id}");
        AutoScalingPolicyReqV11 body = new AutoScalingPolicyReqV11();
        List<String> listExecScriptsNodes = new ArrayList<>();
        listExecScriptsNodes.add("master_node_default_group");
        listExecScriptsNodes.add("core_node_analysis_group");
        listExecScriptsNodes.add("task_node_analysis_group");
        List<String> listExecScriptsNodes1 = new ArrayList<>();
        listExecScriptsNodes1.add("master_node_default_group");
        listExecScriptsNodes1.add("core_node_analysis_group");
        listExecScriptsNodes1.add("task_node_analysis_group");
        List<ScaleScript> listAutoScalingPolicyExecScripts = new ArrayList<>();
        listAutoScalingPolicyExecScripts.add(
            new ScaleScript()
                .withName("before_scale_out")
                .withUri("s3a://XXX/zeppelin_install.sh")
                .withParameters("${mrs_scale_node_num} ${mrs_scale_type} xxx")
                .withNodes(listExecScriptsNodes1)
                .withActiveMaster(true)
                .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
                .withActionStage(ScaleScript.ActionStageEnum.fromValue("before_scale_out"))
        );
        listAutoScalingPolicyExecScripts.add(
            new ScaleScript()
                .withName("after_scale_out")
                .withUri("s3a://XXX/storm_rebalance.sh")
                .withParameters("${mrs_scale_node_hostnames} ${mrs_scale_node_ips}")
                .withNodes(listExecScriptsNodes)
                .withActiveMaster(true)
                .withFailAction(ScaleScript.FailActionEnum.fromValue("continue"))
```



```
            .withActionStage(ScaleScript.ActionStageEnum.fromValue("after_scale_out"))
        );
        Trigger triggerRules = new Trigger();
        triggerRules.withMetricName("YARNMemoryAvailablePercentage")
            .withMetricValue("70")
            .withComparisonOperator("GT")
            .withEvaluationPeriods(10);
        Trigger triggerRules1 = new Trigger();
        triggerRules1.withMetricName("YARNMemoryAvailablePercentage")
            .withMetricValue("25")
            .withComparisonOperator("LT")
            .withEvaluationPeriods(10);
        List<Rule> listAutoScalingPolicyRules = new ArrayList<>();
        listAutoScalingPolicyRules.add(
            new Rule()
                .withName("default-expand-1")
                .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_out"))
                .withCoolDownMinutes(5)
                .withScalingAdjustment(1)
                .withTrigger(triggerRules1)
        );
        listAutoScalingPolicyRules.add(
            new Rule()
                .withName("default-shrink-1")
                .withAdjustmentType(Rule.AdjustmentTypeEnum.fromValue("scale_in"))
                .withCoolDownMinutes(5)
                .withScalingAdjustment(1)
                .withTrigger(triggerRules)
        );
        List<ResourcesPlan> listAutoScalingPolicyResourcesPlans = new ArrayList<>();
        listAutoScalingPolicyResourcesPlans.add(
            new ResourcesPlan()
                .withPeriodType("daily")
                .withStartTime("9:50")
                .withEndTime("10:20")
                .withMinCapacity(2)
                .withMaxCapacity(3)
        );
        listAutoScalingPolicyResourcesPlans.add(
            new ResourcesPlan()
                .withPeriodType("daily")
                .withStartTime("10:20")
                .withEndTime("12:30")
                .withMinCapacity(0)
                .withMaxCapacity(2)
        );
        AutoScalingPolicy autoScalingPolicybody = new AutoScalingPolicy();
        autoScalingPolicybody.withAutoScalingEnable(true)
            .withMinCapacity(1)
            .withMaxCapacity(3)
            .withResourcesPlans(listAutoScalingPolicyResourcesPlans)
            .withRules(listAutoScalingPolicyRules)
            .withExecScripts(listAutoScalingPolicyExecScripts);
        body.withAutoScalingPolicy(autoScalingPolicybody);

body.withNodeGroup(AutoScalingPolicyReqV11.NodeGroupEnum.fromValue("task_node_analysis_group"));
        request.withBody(body);
        try {
            CreateScalingPolicyResponse response = client.createScalingPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
```



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

Python

配置集群弹性伸缩规则

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateScalingPolicyRequest()
        request.cluster_id = "{cluster_id}"
        listNodesExecScripts = [
            "master_node_default_group",
            "core_node_analysis_group",
            "task_node_analysis_group"
        ]
        listNodesExecScripts1 = [
            "master_node_default_group",
            "core_node_analysis_group",
            "task_node_analysis_group"
        ]
        listExecScriptsAutoScalingPolicy = [
            ScaleScript(
                name="before_scale_out",
                uri="s3a://XXX/zeppelin_install.sh",
                parameters="${mrs_scale_node_num} ${mrs_scale_type} xxx",
                nodes=listNodesExecScripts1,
                active_master=True,
                fail_action="continue",
                action_stage="before_scale_out"
            ),
            ScaleScript(
                name="after_scale_out",
                uri="s3a://XXX/storm_rebalance.sh",
                parameters="${mrs_scale_node_hostnames} ${mrs_scale_node_ips}",
                nodes=listNodesExecScripts,
                active_master=True,
                fail_action="continue",
                action_stage="after_scale_out"
            )
        ]
        triggerRules = Trigger(
            metric_name="YARNMemoryAvailablePercentage",
```



```
        metric_value="70",
        comparison_operator="GT",
        evaluation_periods=10
    )
    triggerRules1 = Trigger(
        metric_name="YARNMemoryAvailablePercentage",
        metric_value="25",
        comparison_operator="LT",
        evaluation_periods=10
    )
    listRulesAutoScalingPolicy = [
        Rule(
            name="default-expand-1",
            adjustment_type="scale_out",
            cool_down_minutes=5,
            scaling_adjustment=1,
            trigger=triggerRules1
        ),
        Rule(
            name="default-shrink-1",
            adjustment_type="scale_in",
            cool_down_minutes=5,
            scaling_adjustment=1,
            trigger=triggerRules
        )
    ]
    listResourcesPlansAutoScalingPolicy = [
        ResourcesPlan(
            period_type="daily",
            start_time="9:50",
            end_time="10:20",
            min_capacity=2,
            max_capacity=3
        ),
        ResourcesPlan(
            period_type="daily",
            start_time="10:20",
            end_time="12:30",
            min_capacity=0,
            max_capacity=2
        )
    ]
    autoScalingPolicybody = AutoScalingPolicy(
        auto_scaling_enable=True,
        min_capacity=1,
        max_capacity=3,
        resources_plans=listResourcesPlansAutoScalingPolicy,
        rules=listRulesAutoScalingPolicy,
        exec_scripts=listExecScriptsAutoScalingPolicy
    )
    request.body = AutoScalingPolicyReqV11(
        auto_scaling_policy=autoScalingPolicybody,
        node_group="task_node_analysis_group"
    )
    response = client.create_scaling_policy(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

配置集群弹性伸缩规则

```
package main

import (
```

```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateScalingPolicyRequest{}
    request.ClusterId = "{cluster_id}"
    var listNodesExecScripts = []string{
        "master_node_default_group",
        "core_node_analysis_group",
        "task_node_analysis_group",
    }
    var listNodesExecScripts1 = []string{
        "master_node_default_group",
        "core_node_analysis_group",
        "task_node_analysis_group",
    }
    parametersExecScripts:= "${mrs_scale_node_num} ${mrs_scale_type} xxx"
    activeMasterExecScripts:= true
    parametersExecScripts1:= "${mrs_scale_node_hostnames} ${mrs_scale_node_ips}"
    activeMasterExecScripts1:= true
    var listExecScriptsAutoScalingPolicy = []model.ScaleScript{
        {
            Name: "before_scale_out",
            Uri: "s3a://XXX/zeppelin_install.sh",
            Parameters: &parametersExecScripts,
            Nodes: listNodesExecScripts1,
            ActiveMaster: &activeMasterExecScripts,
            FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
            ActionStage: model.GetScaleScriptActionStageEnum().BEFORE_SCALE_OUT,
        },
        {
            Name: "after_scale_out",
            Uri: "s3a://XXX/storm_rebalance.sh",
            Parameters: &parametersExecScripts1,
            Nodes: listNodesExecScripts,
            ActiveMaster: &activeMasterExecScripts1,
            FailAction: model.GetScaleScriptFailActionEnum().CONTINUE,
            ActionStage: model.GetScaleScriptActionStageEnum().AFTER_SCALE_OUT,
        },
    }
    comparisonOperatorTrigger:= "GT"
    triggerRules := &model.Trigger{
        MetricName: "YARNMemoryAvailablePercentage",
        MetricValue: "70",
```



```
        ComparisonOperator: &comparisonOperatorTrigger,
        EvaluationPeriods: int32(10),
    }
    comparisonOperatorTrigger1:= "LT"
    triggerRules1 := &model.Trigger{
        MetricName: "YARNMemoryAvailablePercentage",
        MetricValue: "25",
        ComparisonOperator: &comparisonOperatorTrigger1,
        EvaluationPeriods: int32(10),
    }
    var listRulesAutoScalingPolicy = []model.Rule{
        {
            Name: "default-expand-1",
            AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_OUT,
            CoolDownMinutes: int32(5),
            ScalingAdjustment: int32(1),
            Trigger: triggerRules1,
        },
        {
            Name: "default-shrink-1",
            AdjustmentType: model.GetRuleAdjustmentTypeEnum().SCALE_IN,
            CoolDownMinutes: int32(5),
            ScalingAdjustment: int32(1),
            Trigger: triggerRules,
        },
    }
    var listResourcesPlansAutoScalingPolicy = []model.ResourcesPlan{
        {
            PeriodType: "daily",
            StartTime: "9:50",
            EndTime: "10:20",
            MinCapacity: int32(2),
            MaxCapacity: int32(3),
        },
        {
            PeriodType: "daily",
            StartTime: "10:20",
            EndTime: "12:30",
            MinCapacity: int32(0),
            MaxCapacity: int32(2),
        },
    }
    autoScalingPolicybody := &model.AutoScalingPolicy{
        AutoScalingEnable: true,
        MinCapacity: int32(1),
        MaxCapacity: int32(3),
        ResourcesPlans: &listResourcesPlansAutoScalingPolicy,
        Rules: &listRulesAutoScalingPolicy,
        ExecScripts: &listExecScriptsAutoScalingPolicy,
    }
    request.Body = &model.AutoScalingPolicyReqV11{
        AutoScalingPolicy: autoScalingPolicybody,
        NodeGroup: model.GetAutoScalingPolicyReqV11NodeGroupEnum().TASK_NODE_ANALYSIS_GROUP,
    }
    response, err := client.CreateScalingPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	操作成功。

错误码

请参见**错误码**。

7.3 标签管理接口

7.3.1 给指定集群添加标签

功能介绍

为特定的集群添加一个tag。

一个集群上最多有20个标签，此接口为幂等接口。添加标签时，如果创建的标签已经存在（key相同），则覆盖。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/clusters/{cluster_id}/tags

表 7-47 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-48 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	**Tag** object	**参数解释：** 集群标签。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-49 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 最大长度128个字符，不能为空字符串。 ● 同一资源的key值不能重复。 ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** ● 最大长度255个字符，可以为空字符串。 ● 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

响应参数

无

请求示例

给指定集群添加标签

```
POST /v1.1/{project_id}/clusters/{cluster_id}/tags

{
  "tag" : {
    "key" : "DEV",
    "value" : "DEV1"
  }
}
```

响应示例

无

状态码

状态码	描述
204	指定集群的标签添加成功

错误码

请参见**错误码**。

7.3.2 查询指定集群的标签

功能介绍

查询指定集群的标签信息。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/clusters/{cluster_id}/tags

表 7-50 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 7-51 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagPlain** objects	**参数解释：** 标签列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-52 TagPlain

参数	参数类型	描述
key	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

请求示例

```
GET /v1.1/{project_id}/clusters/{cluster_id}/tags
```

响应示例

状态码： 200

操作成功。

```
{
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value3"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ListClusterTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListClusterTagsRequest request = new ListClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        try {
            ListClusterTagsResponse response = client.listClusterTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
```



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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        response = client.list_cluster_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```



```
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	操作成功。

错误码

请参见**错误码**。

7.3.3 删除指定集群的标签

功能介绍

删除特定集群的标签。

接口约束

无

调用方法

请参见**如何调用API**。

URI

DELETE /v1.1/{project_id}/clusters/{cluster_id}/tags/{key}

表 7-53 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
key	是	String	**参数解释：** 键。标签的key值。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求参数

无

响应参数

无

请求示例

```
DELETE /v1.1/{project_id}/clusters/{cluster_id}/tags/{key}
```

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.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class DeleteClusterTagSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteClusterTagRequest request = new DeleteClusterTagRequest();
        request.withClusterId("{cluster_id}");
        request.withKey("{key}");
        try {
            DeleteClusterTagResponse response = client.deleteClusterTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteClusterTagRequest()
        request.cluster_id = "{cluster_id}"
        request.key = "{key}"
        response = client.delete_cluster_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
```

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

    request := &model.DeleteClusterTagRequest{}
    request.ClusterId = "{cluster_id}"
    request.Key = "{key}"
    response, err := client.DeleteClusterTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
204	操作成功。

错误码

请参见**错误码**。

7.3.4 批量添加集群标签

功能介绍

为指定集群批量添加标签。

一个集群上最多有20个标签。

此接口为幂等接口：

- 创建时，同一个集群不允许重复key，如果数据库存在就覆盖。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action

表 7-54 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-55 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	**参数解释：** 操作标识：仅限于create（创建）。 **约束限制：** 不涉及 **取值范围：** ● create：创建标签。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
tags	是	Array of **Tag** objects	**参数解释：** 标签列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-56 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** • 最大长度128个字符，不能为空字符串。 • 同一资源的key值不能重复。 • 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** • 最大长度255个字符，可以为空字符串。 • 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

响应参数

无

请求示例

批量添加集群标签

```
POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "DEV1",
    "value" : "DEV1"
  }, {
    "key" : "DEV2",
    "value" : "DEV2"
  } ]
}
```

响应示例

无

状态码

状态码	描述
204	操作成功。

错误码

请参见**错误码**。

7.3.5 批量删除集群标签

功能介绍

为指定集群批量删除标签。

一个集群上最多有20个标签。

此接口为幂等接口：

删除时，如果删除的标签不存在，默认处理成功，删除时不对标签字符集范围做校验。Key长度为128个字符，value为255个字符。删除时tags结构体不能缺失，key不能为空，或者空字符串。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action

表 7-57 路径参数

参数	是否必选	参数类型	描述
cluster_id	是	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-58 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	**参数解释：** 操作标识：仅限于delete（删除）。 **约束限制：** 不涉及 **取值范围：** ● delete：删除标签。 **默认取值：** 不涉及
tags	是	Array of **Tag** objects	**参数解释：** 标签列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-59 Tag

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** ● 最大长度128个字符，不能为空字符串。 ● 同一资源的key值不能重复。 ● 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	是	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** • 最大长度255个字符，可以为空字符串。 • 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

响应参数

无

请求示例

批量删除

```
POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action

{
  "action" : "delete",
  "tags" : [ {
    "key" : "DEV1",
    "value" : "DEV1"
  }, {
    "key" : "DEV2",
    "value" : "DEV2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

批量删除

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

public class BatchDeleteClusterTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchDeleteClusterTagsRequest request = new BatchDeleteClusterTagsRequest();
        request.withClusterId("{cluster_id}");
        BatchDeleteClusterTagsReq body = new BatchDeleteClusterTagsReq();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("DEV1")
                .withValue("DEV1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("DEV2")
                .withValue("DEV2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchDeleteClusterTagsReq.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchDeleteClusterTagsResponse response = client.batchDeleteClusterTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

批量删除

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchDeleteClusterTagsRequest()
        request.cluster_id = "{cluster_id}"
        listTagsbody = [
            Tag(
                key="DEV1",
                value="DEV1"
            ),
            Tag(
                key="DEV2",
                value="DEV2"
            )
        ]
        request.body = BatchDeleteClusterTagsReq(
            tags=listTagsbody,
            action="delete"
        )
        response = client.batch_delete_cluster_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

批量删除

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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



```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteClusterTagsRequest{}
    request.ClusterId = "{cluster_id}"
    var listTagsbody = []model.Tag{
        {
            Key: "DEV1",
            Value: "DEV1",
        },
        {
            Key: "DEV2",
            Value: "DEV2",
        },
    }
    request.Body = &model.BatchDeleteClusterTagsReq{
        Tags: listTagsbody,
        Action: model.GetBatchDeleteClusterTagsReqActionEnum().DELETE,
    }
    response, err := client.BatchDeleteClusterTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
204	操作成功。

错误码

请参见**错误码**。

7.3.6 查询所有标签

功能介绍

查询租户在指定Region下的所有标签集合 。

接口约束

无

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/clusters/tags

表 7-60 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 7-61 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagWithMultiValue** objects	**参数解释：** 标签列表信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-62 TagWithMultiValue

参数	参数类型	描述
key	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
values	Array of strings	**参数解释：** 标签的值。 **约束限制：** 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询所有标签请求示例

```
GET https://{endpoint}/v1.1/{project_id}/clusters/tags
```

响应示例

状态码： 200

操作成功。

```
{
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "value1", "value2" ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ListAllTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAllTagsRequest request = new ListAllTagsRequest();
        try {
            ListAllTagsResponse response = client.listAllTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

if __name__ == "__main__":
```



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

    credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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


更多

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

状态码

状态码	描述
200	操作成功。

错误码

请参见**错误码**。

7.3.7 查询特定标签的集群列表

功能介绍

使用标签过滤集群。

集群默认按照创建时间倒序，集群tag也按照创建时间倒序。

接口约束

无

调用方法

请参见**如何调用API**。

URI

POST /v1.1/{project_id}/clusters/resource_instances/action

表 7-63 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

表 7-64 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of **TagWithMultiValue** objects	**参数解释：** 返回结果包含该参数中所有标签对应的资源。 **约束限制：** 该参数最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。 **取值范围：** 不涉及 **默认取值：** 不涉及
tags_any	否	Array of **TagWithMultiValue** objects	**参数解释：** 返回结果包含该参数中任意一个标签对应的资源。 **约束限制：** 该参数最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
not_tags	否	Array of **TagWithMultiValue** objects	**参数解释：** 返回结果不包含该参数中所有标签对应的资源。 **约束限制：** 该参数最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。 **取值范围：** 不涉及 **默认取值：** 不涉及
not_tags_any	否	Array of **TagWithMultiValue** objects	**参数解释：** 返回结果不包含该参数中任意一个标签对应的资源。 **约束限制：** 该参数最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。 **取值范围：** 不涉及 **默认取值：** 不涉及
action	是	String	**参数解释：** 操作标识（仅限于filter，count），如果是filter则为分页查询，如果是count会按照条件将总条数返回。 **约束限制：** 不涉及 **取值范围：** ● filter：过滤 ● count：查询总条数 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
limit	否	Integer	**参数解释：** 查询记录数。 **约束限制：** action为count时无此参数。如果action为filter默认为1000。 **取值范围：** 1-1000 **默认取值：** 不涉及
offset	否	Integer	**参数解释：** 索引位置，从offset指定的下一条数据开始查询。查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据时响应体中的值带入此参数。 **约束限制：** action为count时无此参数。如果action为filter默认为0。 **取值范围：** ≥0 **默认取值：** 不涉及
matches	否	Array of **Match** objects	**参数解释：** 搜索字段，key为要匹配的字段，如resource_name等。value为匹配的值。此字段为固定字典值。根据不同的字段确认是否需要模糊匹配，如resource_name默认为模糊搜索，如果value为空字符串精确匹配。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-65 TagWithMultiValue

参数	是否必选	参数类型	描述
key	是	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
values	否	Array of strings	**参数解释：** 标签的值。 **约束限制：** 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-66 Match

参数	是否必选	参数类型	描述
key	否	String	**参数解释：** 键。当前只有resource_name可用，表示集群的名称，后续扩展。 **约束限制：** 不涉及 **取值范围：** 长度为[1-64]个字符。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
value	否	String	**参数解释：** 值。每个值最大长度64个unicode字符。 **约束限制：** 不涉及 **取值范围：** 长度为[0-64]个字符。 **默认取值：** 不涉及

响应参数

状态码： 200

表 7-67 响应 Body 参数

参数	参数类型	描述
resources	Array of **MRSResource** objects	**参数解释：** 资源列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
total_count	Integer	**参数解释：** 资源总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-68 MRSResource

参数	参数类型	描述
resource_id	String	**参数解释：** 资源ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_detail	String	**参数解释：** 资源详情。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tags	Array of **TagPlain** objects	**参数解释：** 标签。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_name	String	**参数解释：** 资源名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-69 TagPlain

参数	参数类型	描述
key	String	**参数解释：** 标签的键。 **约束限制：** 不涉及 **取值范围：** 标签的key值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及
value	String	**参数解释：** 标签的值。 **约束限制：** 不涉及 **取值范围：** 标签的value值可以包含任意语种字母、数字、空格和_.:=+-@，但首尾不能含有空格，不能以_sys_开头。 **默认取值：** 不涉及

请求示例

- 查询action为filter时的集群列表

```
POST https://{endpoint}/v1.1/{project_id}/{resource_type}/resource_instances/action

{
  "offset" : "100",
  "limit" : "100",
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "clusterA"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
```

```
 } ]
}
```

- 查询action为count时的集群列表

```
POST https://{endpoint}/v1.1/{project_id}/{resource_type}/resource_instances/action

{
  "action" : "count",
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "matches" : [ {
    "key" : "resource_name",
    "value" : "clusterA"
  } ]
}
```

响应示例

状态码： 200

操作成功。

```
{
  "resources" : [ {
    "resource_detail" : null,
    "resource_id" : "cdfs_cefs_wesas_12_dsad",
    "resource_name" : "clusterA",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    }, {
      "key" : "key2",
      "value" : "value1"
    } ]
  } ],
  "total_count" : "1000"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 查询action为filter时的集群列表

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

public class ListClustersByTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListClustersByTagsRequest request = new ListClustersByTagsRequest();
        ListResourceReq body = new ListResourceReq();
        List<Match> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new Match()
                .withKey("resource_name")
                .withValue("clusterA")
        );
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<TagWithMultiValue> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new TagWithMultiValue()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("value1");
        listNotTagsValues.add("value2");
        List<TagWithMultiValue> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new TagWithMultiValue()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<TagWithMultiValue> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new TagWithMultiValue()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value1");
```

```
        listTagsValues.add("value2");
        List<TagWithMultiValue> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagWithMultiValue()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        body.withMatches(listbodyMatches);
        body.withOffset(100);
        body.withLimit(100);
        body.withAction(ListResourceReq.ActionEnum.fromValue("filter"));
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withNotTags(listbodyNotTags);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            ListClustersByTagsResponse response = client.listClustersByTags(request);
            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());
        }
    }
}
```

- 查询action为count时的集群列表

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;

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

public class ListClustersByTagsSolution {

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

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
```



```
ListClustersByTagsRequest request = new ListClustersByTagsRequest();
ListResourceReq body = new ListResourceReq();
List<Match> listbodyMatches = new ArrayList<>();
listbodyMatches.add(
    new Match()
        .withKey("resource_name")
        .withValue("clusterA")
);
List<String> listNotTagsAnyValues = new ArrayList<>();
listNotTagsAnyValues.add("value1");
listNotTagsAnyValues.add("value2");
List<TagWithMultiValue> listbodyNotTagsAny = new ArrayList<>();
listbodyNotTagsAny.add(
    new TagWithMultiValue()
        .withKey("key1")
        .withValues(listNotTagsAnyValues)
);
List<String> listNotTagsValues = new ArrayList<>();
listNotTagsValues.add("value1");
listNotTagsValues.add("value2");
List<TagWithMultiValue> listbodyNotTags = new ArrayList<>();
listbodyNotTags.add(
    new TagWithMultiValue()
        .withKey("key1")
        .withValues(listNotTagsValues)
);
List<String> listTagsAnyValues = new ArrayList<>();
listTagsAnyValues.add("value1");
listTagsAnyValues.add("value2");
List<TagWithMultiValue> listbodyTagsAny = new ArrayList<>();
listbodyTagsAny.add(
    new TagWithMultiValue()
        .withKey("key1")
        .withValues(listTagsAnyValues)
);
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("value1");
listTagsValues.add("value2");
List<String> listTagsValues1 = new ArrayList<>();
listTagsValues1.add("value1");
listTagsValues1.add("value2");
List<TagWithMultiValue> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new TagWithMultiValue()
        .withKey("key1")
        .withValues(listTagsValues1)
);
listbodyTags.add(
    new TagWithMultiValue()
        .withKey("key2")
        .withValues(listTagsValues)
);
body.withMatches(listbodyMatches);
body.withAction(ListResourceReq.ActionEnum.fromValue("count"));
body.withNotTagsAny(listbodyNotTagsAny);
body.withNotTags(listbodyNotTags);
body.withTagsAny(listbodyTagsAny);
body.withTags(listbodyTags);
request.withBody(body);
try {
    ListClustersByTagsResponse response = client.listClustersByTags(request);
    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

- 查询action为filter时的集群列表

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListClustersByTagsRequest()
        listMatchesbody = [
            Match(
                key="resource_name",
                value="clusterA"
            )
        ]
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesNotTags = [
            "value1",
            "value2"
        ]
        listNotTagsbody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
```



```
            TagWithMultiValue(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesTags
            )
        ]
        request.body = ListResourceReq(
            matches=listMatchesbody,
            offset=100,
            limit=100,
            action="filter",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        response = client.list_clusters_by_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 查询action为count时的集群列表

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListClustersByTagsRequest()
        listMatchesbody = [
            Match(
                key="resource_name",
                value="clusterA"
            )
        ]
        listValuesNotTagsAny = [
            "value1",
```



```
            "value2"
        ]
        listNotTagsAnybody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesNotTags = [
            "value1",
            "value2"
        ]
        listNotTagsbody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listValuesTags1 = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            TagWithMultiValue(
                key="key1",
                values=listValuesTags1
            ),
            TagWithMultiValue(
                key="key2",
                values=listValuesTags
            )
        ]
        request.body = ListResourceReq(
            matches=listMatchesbody,
            action="count",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        response = client.list_clusters_by_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 查询action为filter时的集群列表

```
package main

import (
    "fmt"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClustersByTagsRequest{}
    keyMatches:= "resource_name"
    valueMatches:= "clusterA"
    var listMatchesbody = []model.Match{
        {
            Key: &keyMatches,
            Value: &valueMatches,
        },
    }
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "value1",
        "value2",
    }
    var listNotTagsbody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesNotTags,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
```



```
        "value1",
        "value2",
    }
    var listTagsbody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesTags,
        },
    }
    offsetListResourceReq:= int32(100)
    limitListResourceReq:= int32(100)
    request.Body = &model.ListResourceReq{
        Matches: &listMatchesbody,
        Offset: &offsetListResourceReq,
        Limit: &limitListResourceReq,
        Action: model.GetListResourceReqActionEnum().FILTER,
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    response, err := client.ListClustersByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 查询action为count时的集群列表

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListClustersByTagsRequest{}
    keyMatches:= "resource_name"
    valueMatches:= "clusterA"
    var listMatchesbody = []model.Match{
        {
            Key: &keyMatches,
            Value: &valueMatches,
```



```
        },
    }
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "value1",
        "value2",
    }
    var listNotTagsbody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesNotTags,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    var listValuesTags1 = []string{
        "value1",
        "value2",
    }
    var listTagsbody = []model.TagWithMultiValue{
        {
            Key: "key1",
            Values: &listValuesTags1,
        },
        {
            Key: "key2",
            Values: &listValuesTags,
        },
    }
    request.Body = &model.ListResourceReq{
        Matches: &listMatchesbody,
        Action: model.GetListResourceReqActionEnum().COUNT,
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    response, err := client.ListClustersByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```


更多

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

状态码

状态码	描述
200	操作成功。

错误码

请参见**错误码**。

7.4 可用区

7.4.1 查询可用区信息

功能介绍

在创建集群时，需要配置实例所在的可用区ID，可通过该接口查询可用区的ID。

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{region_id}/available-zones

表 7-70 路径参数

参数	是否必选	参数类型	描述
region_id	是	String	**参数解释：** 区域id，例如cn-north-4。 **约束限制：** 不涉及 **取值范围：** 长度为[1-255]个字符。 **默认取值：** 不涉及

表 7-71 Query 参数

参数	是否必选	参数类型	描述
scope	否	String	**参数解释：** 可用区范围。 **约束限制：** 不涉及 **取值范围：** • Center：表示作用域为中心。 • Edge：表示作用域为边缘。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 7-72 响应 Body 参数

参数	参数类型	描述
available_zones	Array of **AvailableZoneV2** objects	**参数解释：** 可用区列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
default_az_code	String	**参数解释：** 默认可用区编码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
support_physical_az_group	Boolean	**参数解释：** 支持的物理可用区分组。 **约束限制：** 不涉及 **取值范围：** ● True：支持物理可用区分组。 ● False：不支持物理可用区分组。 **默认取值：** 不涉及

表 7-73 AvailableZoneV2

参数	参数类型	描述
id	String	**参数解释：** 可用区编码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
az_code	String	**参数解释：** 可用区编码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
az_name	String	**参数解释：** 可用区名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
az_id	String	**参数解释：** 可用区id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	String	**参数解释：** 可用区状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
region_id	String	**参数解释：** 区域id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
az_group_id	String	**参数解释：** 可用区分组id。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
az_type	String	**参数解释：** 可用区类型。 **约束限制：** 不涉及 **取值范围：** • Core 核心 • Satellite 卫星 • Dedicated 专属 • Virtual 虚拟 • Edge 边缘 • EdgeCental 中心边缘 **默认取值：** 不涉及
az_tags	**AvailableTag** object	**参数解释：** 可用区标签。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-74 AvailableTag

参数	参数类型	描述
mode	String	**参数解释：** 模式，分为专属dedicated和共享shared。 **约束限制：** 不涉及 **取值范围：** • dedicated：专属可用区。 • shared：共享可用区。 **默认取值：** 不涉及

参数	参数类型	描述
alias	String	**参数解释：** az的别名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
public_border_group	String	**参数解释：** 所属group。默认为“center”。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

无

响应示例

状态码： 200

可用区信息

```
{
  "available_zones" : [ {
    "id" : "cn-north-7a",
    "az_code" : "cn-north-7a",
    "az_name" : "可用区1",
    "az_id" : "8c90c2a4e2594c0782faa6b205afeca7",
    "status" : "Running",
    "region_id" : "cn-north-7",
    "az_type" : "Dedicated",
    "az_group_id" : "",
    "az_tags" : {
      "mode" : null,
      "alias" : null,
      "public_border_group" : "center"
    }
  }, {
    "id" : "cn-north-7b",
    "az_code" : "cn-north-7b",
    "az_name" : "可用区2",
    "az_id" : "d539378ec1314c85b76fefa3f7071458",
    "status" : "Running",
    "region_id" : "cn-north-7",
    "az_type" : "Dedicated",
```

```
    "az_tags" : {
      "mode" : null,
      "alias" : null,
      "public_border_group" : "center"
    }
  }, {
    "id" : "cn-north-7c",
    "az_code" : "cn-north-7c",
    "az_name" : "可用区3",
    "az_id" : "9f1c5806706d4c1fb0eb72f0a9b18c77",
    "status" : "Running",
    "region_id" : "cn-north-7",
    "az_type" : "Dedicated",
    "az_tags" : {
      "mode" : null,
      "alias" : null,
      "public_border_group" : "center"
    }
  } ],
  "default_az_code" : "cn-north-7a",
  "support_physical_az_group" : 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.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ListAvailableZonesSolution {

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAvailableZonesRequest request = new ListAvailableZonesRequest();
        request.withRegionId("{region_id}");
        try {
            ListAvailableZonesResponse response = client.listAvailableZones(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListAvailableZonesRequest()
        request.region_id = "{region_id}"
        response = client.list_available_zones(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment 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 := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	可用区信息

错误码

请参见**错误码**。

7.5 版本元数据查询

7.5.1 查询对应版本元数据

功能介绍

查询对应版本元数据。如果参数里指定集群id，则可查询集群更新过补丁之后的最新元数据。

调用方法

请参见**如何调用API**。

URI

GET /v1.1/{project_id}/metadata/versions/{version_name}

表 7-75 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母和数字组成，且长度为[1-64]个字符。 **默认取值：** 不涉及
version_name	是	String	**参数解释：** 集群版本。例如“MRS 3.1.0”。如果请求客户端不支持自动转义，则需要将空格转义为%20，例如“MRS %203.1.0”。 **约束限制：** 不涉及 **取值范围：** 长度为[1-64]个字符。 **默认取值：** 不涉及

表 7-76 Query 参数

参数	是否必选	参数类型	描述
cluster_id	否	String	**参数解释：** 集群ID。如果指定集群ID，则获取该集群做过补丁更新的最新版本元数据。获取方法，请参见**获取集群ID**。 **约束限制：** 不涉及 **取值范围：** 只能由英文字母、数字以及“_”和“-”组成，且长度为[1-64]个字符。 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 7-77 响应 Body 参数

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
name	String	**参数解释：** 镜像版本名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
template_type	String	**参数解释：** 模板类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
image_id	String	**参数解释：** 镜像ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
status	String	**参数解释：** 版本状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
features	Array of strings	**参数解释：** 特性列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_types	Array of strings	**参数解释：** 集群类型列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
version_type	String	**参数解释：** 版本类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
components	Array of **VersionComponent** objects	**参数解释：** 组件列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_requirement	Array of strings	**参数解释：** 版本所需的ip等资源说明。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
constraints	**VersionConstraint** object	**参数解释：** 版本限制说明。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
flavors	FlavorLists object	**参数解释：** 规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
role_deploy_meta	Array of RoleDeployMeta objects	**参数解释：** 版本组件实例角色部署策略。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-78 VersionComponent

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
name	String	**参数解释：** 组件名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
version	String	**参数解释：** 支持版本。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
depend_on	Array of strings	**参数解释：** 组件依赖项。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
description	String	**参数解释：** 组件描述。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
available_cluster_types	Array of strings	**参数解释：** 支持该组件的集群类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
external_datasources	Array of **ComponentExternalDatasource** objects	**参数解释：** 外部数据源。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
resource_requirement	Array of strings	**参数解释：** 所需的ip等资源说明。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
valid_roles	Array of strings	**参数解释：** 有效角色。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
visible	Boolean	**参数解释：** 是否可见。 **约束限制：** 不涉及 **取值范围：** ● True：可见。 ● False：不可见。 **默认取值：** 不涉及

参数	参数类型	描述
children_components	Array of strings	**参数解释：** 子组件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
multi_az_support_status	String	**参数解释：** 多az支持状态。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-79 ComponentExternalDatasource

参数	参数类型	描述
name	String	**参数解释：** 外部数据源名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
types	Array of strings	**参数解释：** 外部数据源类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-80 VersionConstraint

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_constraint	**NodeConstraints** object	**参数解释：** 节点限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
safe_mode_kerberos_exclude_components	Array of strings	**参数解释：** 安全模式kerberos排除组件列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-81 NodeConstraints

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他节点限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
master	NodeConstraint object	**参数解释：** master节点限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
core	NodeConstraint object	**参数解释：** core节点限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task	NodeConstraint object	**参数解释：** task节点限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
core_separate	NodeConstraint object	**参数解释：** core节点管控分离限制，管控分离模板下core节点的限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
core_combine	**NodeConstraint** object	**参数解释：** core节点管控合设限制，管控合设模板下core节点的限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task_separate	**NodeConstraint** object	**参数解释：** task节点管控分离限制，管控分离模板下task节点的限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task_combine	**NodeConstraint** object	**参数解释：** task节点管控合设限制，管控合设模板下task节点的限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_group_task	**NodeConstraint** object	**参数解释：** task节点组限制，包括节点数，磁盘等限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-82 NodeConstraint

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
min_node_num	Integer	**参数解释：** 最少节点数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
max_node_num	Integer	**参数解释：** 最多节点数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
min_core_num	Map<String,Integer>	**参数解释：** 最少核心数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
min_mem_size	Map<String,Integer>	**参数解释：** 最小内存容量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
min_disk_size	Integer	**参数解释：** 最小磁盘容量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
max_node_group_num	Integer	**参数解释：** 最大节点组数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
min_data_volume_total_size	Map<String,Integer>	**参数解释：** 最小数据卷容量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
disk_type_constraint	Map<String,String>	**参数解释：** 磁盘类型限制，包含当前节点组所支持的磁盘类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
min_root_disk_size	Integer	**参数解释：** 最小系统磁盘大小。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-83 FlavorLists

参数	参数类型	描述
master	Array of strings	**参数解释：** master节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
core	Array of strings	**参数解释：** core节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
task	Array of strings	**参数解释：** task节点支持的规格列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 7-84 RoleDeployMeta

参数	参数类型	描述
other	Map<String,Object>	**参数解释：** 其他扩展属性。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
name	String	**参数解释：** 角色名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
code_name	String	**参数解释：** 角色简称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
component	String	**参数解释：** 角色所属组件。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
node_preference	String	**参数解释：** 部署倾向。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
count	String	**参数解释：** 角色数量限制。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
affinity	String	**参数解释：** 亲和。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
affinity_target	String	**参数解释：** 亲和目标。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
multi_instance	Integer	**参数解释：** 多实例。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
role_kind	String	**参数解释：** 角色类型。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
constraints	Array of strings	**参数解释：** 角色限制，包含当前组件角色的一些功能限制，例如："no_scale_in"。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
multi_az_placement	String	**参数解释：** 多az部署。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
arbitration_deployment	Boolean	**参数解释：** 仲裁部署。 **约束限制：** 不涉及 **取值范围：** ● true：存在仲裁部署。 ● false：不存在仲裁部署。 **默认取值：** 不涉及
support_elb	Boolean	**参数解释：** 支持ELB。 **约束限制：** 不涉及 **取值范围：** ● true：支持ELB。 ● false：不支持ELB。 **默认取值：** 不涉及
multi_affinity_group_enable	Boolean	**参数解释：** 启用多亲和组。 **约束限制：** 不涉及 **取值范围：** ● true：启用多亲和组。 ● false：不启用多亲和组。 **默认取值：** 不涉及

参数	参数类型	描述
local_disks_anti_affinity	Boolean	**参数解释：** 本地盘反亲和。 **约束限制：** 不涉及 **取值范围：** ● true：本地盘反亲和。 ● false：本地盘反不亲和。 **默认取值：** 不涉及
multi_instance_name_pattern	String	**参数解释：** 多实例名称模式。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
private_ip	String	**参数解释：** 私有IP。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
weight	String	**参数解释：** 权重。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

获取MRS 3.1.0的版本元数据

```
GET /v1.1/{project_id}/metadata/versions/MRS%203.1.0
```

响应示例

状态码： 200

版本元数据详情。（该样例省略了部分内容，详细返回请调用接口查看）


```
{
  "other" : { },
  "name" : "MRS 3.1.0",
  "template_type" : "clusterFusion",
  "image_id" : "ECS:X86:deda7c89-6069-46d3-af7c-4afc31eec0d9,ECS:ARM:d5f0087b-9fd7-4977-b9c8-
d07262f452d6,BMS:ks1ne:bf071a00-d1d6-472f-943b-22f733248f36,BMS:d2:a9a1a279-85c9-4c05-
bd0a-4cb786937a09",
  "status" : "GA",
  "features" : [ "anti_affinity", "auto_scaling", "bootstrap_scripts", "bootstrap_support_obs", "custom_config",
"custom_topology", "datasource_manager", "decouple_install", "elastic_ip", "fi_cluster", "fi_cluster_v2",
"iam_user_sync", "log_collect", "log_collect_v2", "master_scale_up", "mrs_ecs_agency", "multi_disks",
"multi_login_mode", "obs_integration", "obs_user_policy", "opensource_port_matrix", "ops_channel",
"password_policy_v2", "random_passwd_for_install", "register_dns_server", "resize", "resources_plans",
"safe_mode", "scale_scripts", "security_job_submit", "smn_alarm", "spark_sql_validation_in_executor",
"support_obs_protocol", "task_node", "unified_management_ui", "sql_execution", "multi_az",
"weak_multi_az_deployment_constraint", "multi_master_scale_up", "omm_random_pwd", "metric_monitor",
"auth_manager", "cluster_patch", "hw_domain_name", "fi_custom_config", "force_scale_up",
"skip_fi_evs_expand", "detach_node" ],
  "cluster_types" : [ "analysis", "streaming", "mixed", "custom" ],
  "version_type" : "basic",
  "components" : [ {
    "other" : { },
    "name" : "Hadoop",
    "version" : "3.1.1",
    "depend_on" : [ "ZooKeeper", "Ranger" ],
    "description" : "A framework that allows for the distributed processing of large data sets across clusters.",
    "available_cluster_types" : [ "analysis", "mixed", "custom" ],
    "external_datasources" : null,
    "resource_requirement" : [ "privateIp:1" ],
    "valid_roles" : [ "NameNode", "Zkfc", "JournalNode", "DataNode", "ResourceManager", "NodeManager",
"JobHistoryServer", "TimelineServer", "HttpFS" ],
    "visible" : true,
    "children_components" : [ "HDFS", "Mapreduce", "Yarn" ],
    "multi_az_support_status" : "multi_az_ha"
  }, {
    "other" : { },
    "name" : "Spark2x",
    "version" : "2.4.5",
    "depend_on" : [ "HDFS", "Yarn", "Hive", "KrbClient", "KrbServer", "ZooKeeper" ],
    "description" : "Apache Spark2x is a fast and general engine based on open source Spark2.x for large-
scale data processing.",
    "available_cluster_types" : [ "analysis", "mixed", "custom" ],
    "external_datasources" : null,
    "resource_requirement" : null,
    "valid_roles" : [ "JobHistory2x", "JDBCServer2x", "SparkResource2x", "IndexServer2x" ],
    "visible" : true,
    "children_components" : [ "Spark2x" ],
    "multi_az_support_status" : "multi_az_ha"
  }, {
    "other" : { },
    "name" : "HBase",
    "version" : "2.2.3",
    "depend_on" : [ "HDFS", "Yarn", "KrbServer", "ZooKeeper" ],
    "description" : "A scalable, distributed database that supports structured data storage for large tables.",
    "available_cluster_types" : [ "analysis", "mixed", "custom" ],
    "external_datasources" : null,
    "resource_requirement" : null,
    "valid_roles" : [ "HMaster", "RegionServer", "ThriftServer", "Thrift1Server", "RESTServer",
"RegionServer_1" ],
    "visible" : true,
    "children_components" : [ "HBase" ],
    "multi_az_support_status" : "multi_az_ha"
  }, {
```



```
  "other" : { },
  "name" : "Hive",
  "version" : "3.1.0",
  "depend_on" : [ "HDFS", "Mapreduce", "Yarn", "DBService", "ZooKeeper" ],
  "description" : "A data warehouse infrastructure that provides data summarization and ad hoc querying.",
  "available_cluster_types" : [ "analysis", "mixed", "custom" ],
  "external_datasources" : [ {
    "name" : "hive_metastore",
    "types" : [ "RDS_MYSQL", "DLCATALOG" ]
  } ],
  "resource_requirement" : null,
  "valid_roles" : [ "MetaStore", "WebHCat", "HiveServer" ],
  "visible" : true,
  "children_components" : [ "Hive" ],
  "multi_az_support_status" : "multi_az_ha"
}, {
  "other" : { },
  "name" : "ZooKeeper",
  "version" : null,
  "depend_on" : [ "KrbClient", "KrbServer" ],
  "description" : null,
  "available_cluster_types" : [ "analysis", "streaming", "mixed", "custom" ],
  "external_datasources" : null,
  "resource_requirement" : null,
  "valid_roles" : [ "quorumpeer" ],
  "visible" : true,
  "children_components" : [ "ZooKeeper" ],
  "multi_az_support_status" : "multi_az_ha"
} ],
"resource_requirement" : [ "privateIp:2" ],
"constraints" : {
  "other" : { },
  "node_constraint" : {
    "other" : { },
    "master" : {
      "other" : { },
      "min_node_num" : 2,
      "max_node_num" : 9,
      "min_core_num" : null,
      "min_mem_size" : null,
      "min_disk_size" : 600,
      "max_node_group_num" : 1,
      "min_data_volume_total_size" : null,
      "disk_type_constraint" : null,
      "min_root_disk_size" : 480
    },
    "core" : {
      "other" : { },
      "min_node_num" : 3,
      "max_node_num" : 500,
      "min_core_num" : null,
      "min_mem_size" : null,
      "min_disk_size" : 600,
      "max_node_group_num" : null,
      "min_data_volume_total_size" : null,
      "disk_type_constraint" : null,
      "min_root_disk_size" : 480
    },
    "task" : null,
    "core_separate" : null,
    "core_combine" : null,
    "task_separate" : null,
    "task_combine" : null,
    "node_group_task" : {
      "other" : { },
      "min_node_num" : 2,
      "max_node_num" : 10000,
      "min_core_num" : null,
      "min_mem_size" : null,
```



```
      "min_disk_size" : 600,
      "max_node_group_num" : 9,
      "min_data_volume_total_size" : null,
      "disk_type_constraint" : null,
      "min_root_disk_size" : 480
    }
   },
   "safe_mode_kerberos_exclude_components" : [ "Presto", "Pulsar" ]
  },
  "flavors" : {
    "master" : [ "ac7.4xlarge.4", "ac7.8xlarge.4", "ac7.16xlarge.4", "ac7.32xlarge.4", "ac7.8xlarge.2",
"ac7.16xlarge.2", "ac7.32xlarge.2", "am7.2xlarge.8", "am7.4xlarge.8", "am7.8xlarge.8", "c3ne.4xlarge.4",
"c3ne.8xlarge.4", "c3ne.15xlarge.4" ],
    "core" : [ "ac7.4xlarge.4", "ac7.8xlarge.4", "ac7.16xlarge.4", "ac7.32xlarge.4", "ac7.8xlarge.2",
"ac7.16xlarge.2", "ac7.32xlarge.2", "am7.2xlarge.8", "am7.4xlarge.8", "am7.8xlarge.8", "c3ne.4xlarge.4",
"c3ne.8xlarge.4", "c3ne.15xlarge.4", "c6.4xlarge.4", "c6.8xlarge.2", "c6.8xlarge.4", "c6.16xlarge.2" ],
    "task" : [ "ac7.4xlarge.4", "ac7.8xlarge.4", "ac7.16xlarge.4", "ac7.32xlarge.4", "ac7.8xlarge.2",
"ac7.16xlarge.2", "ac7.32xlarge.2", "am7.2xlarge.8", "am7.4xlarge.8", "am7.8xlarge.8", "c3ne.4xlarge.4",
"c3ne.8xlarge.4", "c3ne.15xlarge.4" ]
  },
  "role_deploy_meta" : [ {
    "other" : { },
    "name" : "OMSServer",
    "code_name" : "OMS",
    "component" : "OMSServer",
    "node_preference" : "MASTER",
    "count" : "2",
    "affinity" : null,
    "affinity_target" : null,
    "multi_instance" : null,
    "role_kind" : null,
    "constraints" : null,
    "multi_az_placement" : "AT_LEAST_2",
    "arbitration_deployment" : false,
    "support_elb" : false,
    "multi_affinity_group_enable" : false,
    "local_disks_anti_affinity" : false,
    "multi_instance_name_pattern" : null,
    "private_ip" : null,
    "weight" : null
  }, {
    "other" : { },
    "name" : "NameNode",
    "code_name" : "NN",
    "component" : "HDFS",
    "node_preference" : "MASTER",
    "count" : "2",
    "affinity" : null,
    "affinity_target" : null,
    "multi_instance" : null,
    "role_kind" : null,
    "constraints" : null,
    "multi_az_placement" : "AT_LEAST_2",
    "arbitration_deployment" : false,
    "support_elb" : false,
    "multi_affinity_group_enable" : false,
    "local_disks_anti_affinity" : false,
    "multi_instance_name_pattern" : null,
    "private_ip" : null,
    "weight" : null
  }, {
    "other" : { },
    "name" : "Zkfc",
    "code_name" : "ZKFC",
    "component" : "HDFS",
    "node_preference" : null,
    "count" : null,
    "affinity" : "JUST_COLOCATE",
    "affinity_target" : "NameNode",
```



```
  "multi_instance" : null,
  "role_kind" : null,
  "constraints" : null,
  "multi_az_placement" : "AT_LEAST_2",
  "arbitration_deployment" : false,
  "support_elb" : false,
  "multi_affinity_group_enable" : false,
  "local_disks_anti_affinity" : false,
  "multi_instance_name_pattern" : null,
  "private_ip" : null,
  "weight" : null
}, {
  "other" : { },
  "name" : "HttpFS",
  "code_name" : "HFS",
  "component" : "HDFS",
  "node_preference" : "MASTER",
  "count" : "[0-10]",
  "affinity" : null,
  "affinity_target" : null,
  "multi_instance" : null,
  "role_kind" : null,
  "constraints" : null,
  "multi_az_placement" : "AT_LEAST_2",
  "arbitration_deployment" : false,
  "support_elb" : false,
  "multi_affinity_group_enable" : false,
  "local_disks_anti_affinity" : false,
  "multi_instance_name_pattern" : null,
  "private_ip" : null,
  "weight" : null
}, {
  "other" : { },
  "name" : "JournalNode",
  "code_name" : "JN",
  "component" : "HDFS",
  "node_preference" : "MASTER",
  "count" : "[3-60],step=2",
  "affinity" : null,
  "affinity_target" : null,
  "multi_instance" : null,
  "role_kind" : null,
  "constraints" : null,
  "multi_az_placement" : "QUORUM_LIKE",
  "arbitration_deployment" : true,
  "support_elb" : false,
  "multi_affinity_group_enable" : false,
  "local_disks_anti_affinity" : false,
  "multi_instance_name_pattern" : null,
  "private_ip" : null,
  "weight" : null
}, {
  "other" : { },
  "name" : "DataNode",
  "code_name" : "DN",
  "component" : "HDFS",
  "node_preference" : "NO_LIMIT",
  "count" : "[3-10000]",
  "affinity" : null,
  "affinity_target" : null,
  "multi_instance" : null,
  "role_kind" : "stateful",
  "constraints" : null,
  "multi_az_placement" : "AT_LEAST_2",
  "arbitration_deployment" : false,
  "support_elb" : false,
  "multi_affinity_group_enable" : false,
  "local_disks_anti_affinity" : false,
  "multi_instance_name_pattern" : null,
```



```
    "private_ip" : null,
    "weight" : null
  }, {
    "other" : { },
    "name" : "ResourceManager",
    "code_name" : "RM",
    "component" : "Yarn",
    "node_preference" : "MASTER",
    "count" : "2",
    "affinity" : null,
    "affinity_target" : null,
    "multi_instance" : null,
    "role_kind" : null,
    "constraints" : null,
    "multi_az_placement" : "AT_LEAST_2",
    "arbitration_deployment" : false,
    "support_elb" : false,
    "multi_affinity_group_enable" : false,
    "local_disks_anti_affinity" : false,
    "multi_instance_name_pattern" : null,
    "private_ip" : null,
    "weight" : null
  }, {
    "other" : { },
    "name" : "NodeManager",
    "code_name" : "NM",
    "component" : "Yarn",
    "node_preference" : "NO_LIMIT",
    "count" : "[3-10000]",
    "affinity" : null,
    "affinity_target" : null,
    "multi_instance" : null,
    "role_kind" : "stateless",
    "constraints" : null,
    "multi_az_placement" : "AT_LEAST_2",
    "arbitration_deployment" : false,
    "support_elb" : false,
    "multi_affinity_group_enable" : false,
    "local_disks_anti_affinity" : false,
    "multi_instance_name_pattern" : null,
    "private_ip" : null,
    "weight" : 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.mrs.v1.region.MrsRegion;
import com.huaweicloud.sdk.mrs.v1.*;
import com.huaweicloud.sdk.mrs.v1.model.*;


public class ShowMrsVersionMetadataSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

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

        MrsClient client = MrsClient.newBuilder()
                .withCredential(auth)
                .withRegion(MrsRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowMrsVersionMetadataRequest request = new ShowMrsVersionMetadataRequest();
        request.withVersionName("{version_name}");
        try {
            ShowMrsVersionMetadataResponse response = client.showMrsVersionMetadata(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkmrs.v1.region.mrs_region import MrsRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkmrs.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowMrsVersionMetadataRequest()
        request.version_name = "{version_name}"
        response = client.show_mrs_version_metadata(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```



```
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    mrs "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/mrs/v1/region"
)

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

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

    client := mrs.NewMrsClient(
        mrs.MrsClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	版本元数据详情。（该样例省略了部分内容，详细返回请调用接口查看）

错误码

请参见**错误码**。

8 历史 API

8.1 V1.1 作业管理接口（废弃）

8.1.1 新增作业并执行（废弃）

功能介绍

在MRS集群中新增一个作业，并执行作业。该接口不兼容Sahara。 集群ID可参考**查询集群列表**接口获取。

说明

MRS 3.x版本镜像，不支持MRS V1.1作业管理接口，需要使用**V2作业管理接口**。

接口约束

- DistCp作业需要配置文件操作类型(file_action)。
- Spark SQL需要配置Spark SQL语句(hql)和sql程序路径(hive_script_path)。
- 开启Kerberos认证的安全集群暂不支持使用该接口提交作业。

调试

您可以在**API Explorer**中调试该接口。

URI

POST /v1.1/{project_id}/jobs/submit-job

表 8-1 路径参数

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

请求参数

表 8-2 请求 Body 参数

参数	是否必选	参数类型	描述
job_name	是	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由字母、数字、中划线和下划线组成，并且长度为1～64个字符。 不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
jar_path	否	String	**参数解释：** 执行程序Jar包或sql文件地址。 **约束限制：** 作业类型为MapReduce或Spark时，jar_path参数为必选。 **取值范围：** ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 需要以“/”或“s3a://”开头。OBS路径不支持KMS加密的文件或程序。 ● Spark Script需要以“.sql”结尾，MapReduce和Spark Jar需要以“.jar”结尾，sql和jar不区分大小写。 **默认取值：** 不涉及
input	否	String	**参数解释：** 数据输入地址。 **约束限制：** 不涉及 **取值范围：** 必须以“/”或“s3a://”开头。请配置为正确的OBS路径，OBS路径不支持KMS加密的文件或程序。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
output	否	String	**参数解释：** 数据输出地址。 **约束限制：** 不涉及 **取值范围：** 必须以“/”或“s3a://”开头。请配置为正确的OBS路径，如果该路径不存在，系统会自动创建。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
job_log	否	String	**参数解释：** 作业日志存储地址，该日志信息记录作业运行状态。 **约束限制：** 不涉及 **取值范围：** 必须以“/”或“s3a://”开头，请配置为正确的OBS路径。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
job_type	是	Integer	**参数解释：** 作业类型码。 **约束限制：** 不涉及 **取值范围：** ● 1：MapReduce ● 2：Spark ● 3：Hive Script ● 4：HiveSQL（当前不支持） ● 5：DistCp，导入、导出数据。 ● 6：Spark Script ● 7：Spark SQL，提交SQL语句（该接口当前不支持） **默认取值：** 不涉及
file_action	否	String	**参数解释：** 文件操作类型。 **约束限制：** 不涉及 **取值范围：** ● export：从HDFS导出数据至OBS。 ● import：从OBS导入数据至HDFS。 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
arguments	否	String	**参数解释：** 程序执行的关键参数。 **约束限制：** 该参数由用户程序内的函数指定，MRS只负责参数的传入。 **取值范围：** 最多为150000字符，不能包含;|&>'<$!'"\特殊字符，可为空。 说明： 用户输入带有敏感信息（如登录密码）的参数时，可通过在参数名前添加“@”的方式，为该参数值加密，以防止敏感信息被明文形式持久化。在查看作业信息时，敏感信息显示为“*”。 例如：username=xxx @password=yyy **默认取值：** 不涉及

参数	是否必选	参数类型	描述
hql	否	String	**参数解释：** Spark SQL语句。 **约束限制：** 该语句需要进行Base64编码和解码，“ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/”为标准的编码表，MRS使用“ABCDEFGHILKJMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/”进行Base64编码。在编码后所得字符串首位任意加上一个字母，即得到Hql参数的值。后台自动进行解码得到Spark SQL语句。 使用样例： 1. 在Web界面输入Spark SQL语句“show tables;”。 2. 使用“ABCDEFGHILKJMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/”编码后得到字符串“c2hvdyB0YWLsZXM7”。 3. 在“c2hvdyB0YWLsZXM7”首位任意加上一字母，例如“gc2hvdyB0YWLsZXM7”，即Hql参数的值。 4. 后台自动进行解码得到Spark SQL语句“show tables;”。 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
hive_script_path	否	String	**参数解释：** sql程序路径。 **约束限制：** 仅Spark Script和Hive Script作业需要使用此参数。 **取值范围：** ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 需要以“/”或“s3a://”开头，OBS路径不支持KMS加密的文件或程序。 ● 需要以“.sql”结尾，sql不区分大小写。 **默认取值：** 不涉及

响应参数

状态码： 200

表 8-3 响应 Body 参数

参数	参数类型	描述
templated	Boolean	**参数解释：** 作业执行对象是否由作业模板生成。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
created_at	Long	**参数解释：** 作业创建时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
updated_at	Long	**参数解释：** 作业更新时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenant_id	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_id	String	**参数解释：** 作业应用ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_name	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 只能由字母、数字、中划线和下划线组成，并且长度为1～64个字符。 不同作业的名称允许相同，但不建议设置相同。 **默认取值：** 不涉及
input_id	String	**参数解释：** 数据输入ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
output_id	String	**参数解释：** 数据输出ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
start_time	Long	**参数解释：** 作业执行开始时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	Long	**参数解释：** 作业执行结束时间，十位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_id	String	**参数解释：** 集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
engine_job_id	String	**参数解释：** Oozie工作流ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
return_code	String	**参数解释：** 运行结果返回码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
is_public	Boolean	**参数解释：** 是否公开。 **约束限制：** 当前版本不支持该功能。 **取值范围：** 不涉及 **默认取值：** 不涉及
is_protected	Boolean	**参数解释：** 是否受保护。 **约束限制：** 当前版本不支持该功能。 **取值范围：** 不涉及 **默认取值：** 不涉及
group_id	String	**参数解释：** 作业执行组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
jar_path	String	**参数解释：** 执行程序Jar包或sql文件地址。 **约束限制：** 不涉及 **取值范围：** ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 需要以“/”或“s3a://”开头。OBS路径不支持KMS加密的文件或程序。 ● Spark Script需要以“.sql”结尾，MapReduce和Spark Jar需要以“.jar”结尾，sql和jar不区分大小写。 **默认取值：** 不涉及
input	String	**参数解释：** 数据输入地址。 **约束限制：** 请配置为正确的OBS路径，OBS路径不支持KMS加密的文件或程序。 **取值范围：** 必须以“/”或“s3a://”开头。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
output	String	**参数解释：** 数据输出地址。 **约束限制：** 不涉及 **取值范围：** 必须以“/”或“s3a://”开头。请配置为正确的OBS路径，如果该路径不存在，系统会自动创建。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及

参数	参数类型	描述
job_log	String	**参数解释：** 作业日志存储地址，该日志信息记录作业运行状态。 **约束限制：** 不涉及 **取值范围：** 必须以“/”或“s3a://”开头，请配置为正确的OBS路径。 最多为1023字符，不能包含;|&>'<$特殊字符，可为空。 **默认取值：** 不涉及
job_type	Integer	**参数解释：** 作业类型码。 **约束限制：** 只有包含Spark和Hive组件的集群才能新增Spark和Hive类型的作业。 **取值范围：** ● 1：MapReduce ● 2：Spark ● 3：Hive Script ● 4：HiveSQL（当前不支持） ● 5：DistCp，导入、导出数据。 ● 6：Spark Script ● 7：Spark SQL，提交SQL语句，（该接口当前不支持） **默认取值：** 不涉及
file_action	String	**参数解释：** 文件操作类型。 **约束限制：** 不涉及 **取值范围：** ● export：从HDFS导出数据至OBS。 ● import：从OBS导入数据至HDFS。 **默认取值：** 不涉及

参数	参数类型	描述
arguments	String	**参数解释：** 程序执行的关键参数。 **约束限制：** 该参数由用户程序内的函数指定，MRS只负责参数的传入。 **取值范围：** 最多为150000字符，不能包含;|&>'<$!"\特殊字符，可为空。 说明： 用户输入带有敏感信息（如登录密码）的参数时，可通过在参数名前添加“@”的方式，为该参数值加密，以防止敏感信息被明文形式持久化。在查看作业信息时，敏感信息显示为“*”。 例如：username=admin @password=*** **默认取值：** 不涉及
hql	String	**参数解释：** Hive&Spark Sql语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_state	Integer	**参数解释：** 作业状态码。 **约束限制：** 不涉及 **取值范围：** ● -1：Terminated ● 1：Starting ● 2：Running ● 3：Completed ● 4：Abnormal ● 5：Error **默认取值：** 不涉及

参数	参数类型	描述
job_final_status	Integer	**参数解释：** 作业最终状态码。 **约束限制：** 不涉及 **取值范围：** ● 0：未完成 ● 1：执行错误，终止执行 ● 2：执行完成并且成功 ● 3：已取消 **默认取值：** 不涉及
hive_script_path	String	**参数解释：** sql程序路径。 **约束限制：** 仅Spark Script和Hive Script作业需要使用此参数。 **取值范围：** ● 最多为1023字符，不能包含;|&><'$特殊字符，且不可为空或全空格。 ● 需要以“/”或“s3a://”开头，OBS路径不支持KMS加密的文件或程序。 ● 需要以“.sql”结尾，sql不区分大小写。 **默认取值：** 不涉及
create_by	String	**参数解释：** 创建作业的用户ID。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
finished_step	Integer	**参数解释：** 当前已完成的步骤数。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_main_id	String	**参数解释：** 作业主ID。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_step_id	String	**参数解释：** 作业步骤ID。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
postpone_at	Long	**参数解释：** 延迟时间，十位时间戳。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
step_name	String	**参数解释：** 作业步骤名。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
step_num	Integer	**参数解释：** 步骤数量 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task_num	Integer	**参数解释：** 任务数量。 为兼容历史版本，保留此参数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
update_by	String	**参数解释：** 更新作业的用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
credentials	String	**参数解释：** 令牌，当前版本不支持。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
user_id	String	**参数解释：** 创建作业的用户ID。 历史版本兼容，不再使用。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_configs	Map<String,Object>	**参数解释：** 键值对集合，用于保存作业运行配置。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
extra	Map<String,Object>	**参数解释：** 认证信息，当前版本不支持。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
data_source_urls	Map<String,Object>	**参数解释：** 数据源URL。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
info	Map<String,Object>	**参数解释：** 键值对集合，包含oozie返回的作业运行信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

- MapReduce作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 1,
  "job_name" : "mrs_test_jobone_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "jar_path" : "s3a://mrs-opsadm/jarpath/hadoop-mapreduce-examples-2.7.2.jar",
  "arguments" : "wordcount",
  "input" : "s3a://mrs-opsadm/input/",
  "output" : "s3a://mrs-opsadm/output/",
  "job_log" : "s3a://mrs-opsadm/log/",
  "file_action" : "",
  "hql" : "",
  "hive_script_path" : ""
}
```

- Spark作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 2,
  "job_name" : "mrs_test_sparkjob_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "jar_path" : "s3a://mrs-opsadm/jarpath/spark-test.jar",
  "arguments" : "org.apache.spark.examples.SparkPi 10",
  "input" : "",
  "output" : "s3a://mrs-opsadm/output/",
  "job_log" : "s3a://mrs-opsadm/log/",
  "file_action" : "",
  "hql" : "",
  "hive_script_path" : ""
}
```

- Hive Script作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 3,
  "job_name" : "mrs_test_SparkScriptJob_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "jar_path" : "s3a://mrs-opsadm/jarpath/Hivescript.sql",
  "arguments" : "",
  "input" : "s3a://mrs-opsadm/input/",
  "output" : "s3a://mrs-opsadm/output/",
  "job_log" : "s3a://mrs-opsadm/log/",
  "file_action" : "",
  "hql" : "",
  "hive_script_path" : "s3a://mrs-opsadm/jarpath/Hivescript.sql"
}
```

- DistCp导入作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 5,
  "job_name" : "mrs_test_importjob_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "input" : "s3a://mrs-opsadm/jarpath/hadoop-mapreduce-examples-2.7.2.jar",
  "output" : "/user",
  "file_action" : "import"
}
```

- DistCp导出作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 5,
  "job_name" : "mrs_test_exportjob_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "input" : "/user/hadoop-mapreduce-examples-2.7.2.jar",
  "output" : "s3a://mrs-opsadm/jarpath/",
  "file_action" : "export"
}
```

- Spark Script作业请求示例

```
POST https://{endpoint}/v1.1/{project_id}/jobs/submit-job

{
  "job_type" : 6,
  "job_name" : "mrs_test_sparkscriptjob_20170602_141106",
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "jar_path" : "s3a://mrs-opsadm/jarpath/sparkscript.sql",
  "arguments" : "",
  "input" : "s3a://mrs-opsadm/input/",
  "output" : "s3a://mrs-opsadm/output/",
  "job_log" : "s3a://mrs-opsadm/log/",
  "file_action" : "",
  "hql" : "",
  "hive_script_path" : "s3a://mrs-opsadm/jarpath/sparkscript.sql"
}
```

响应示例

状态码： 200

新增作业成功。

```
{
  "job_execution" : {
    "templated" : "false",
    "created_at" : "1496387588",
    "updated_at" : "1496387588",
```



```
  "id" : "12ee9ae4-6ee1-48c6-bb84-fb0b4f76cf03",
  "tenant_id" : "c71ad83a66c5470496c2ed6e982621cc",
  "job_id" : "",
  "job_name" : "mrs_test_jobone_20170602_141106",
  "input_id" : null,
  "output_id" : null,
  "start_time" : "1496387588",
  "end_time" : null,
  "cluster_id" : "e955a7a3-d334-4943-a39a-994976900d56",
  "engine_job_id" : null,
  "return_code" : null,
  "is_public" : null,
  "is_protected" : null,
  "group_id" : "12ee9ae4-6ee1-48c6-bb84-fb0b4f76cf03",
  "jar_path" : "s3a://mrs-opsadm/jarpath/hadoop-mapreduce-examples-2.7.2.jar",
  "input" : "s3a://mrs-opsadm/input/",
  "output" : "s3a://mrs-opsadm/output/",
  "job_log" : "s3a://mrs-opsadm/log/",
  "job_type" : "1",
  "file_action" : "",
  "arguments" : "wordcount",
  "hql" : "",
  "job_state" : "2",
  "job_final_status" : "0",
  "hive_script_path" : "",
  "create_by" : "b67132be2f054a45b247365647e05af0",
  "finished_step" : "0",
  "job_main_id" : "",
  "job_step_id" : "",
  "postpone_at" : "1496387588",
  "step_name" : "",
  "step_num" : "0",
  "task_num" : "0",
  "update_by" : "b67132be2f054a45b247365647e05af0",
  "credentials" : "",
  "user_id" : "b67132be2f054a45b247365647e05af0",
  "job_configs" : null,
  "extra" : null,
  "data_source_urls" : null,
  "info" : null
 }
}
```


状态码

状态码	描述
200	新增作业成功。

错误码

请参见**错误码**。

8.1.2 查询作业 exe 对象列表（废弃）

功能介绍

查询所有作业的exe对象列表。该接口不兼容Sahara。

说明

MRS 3.x版本镜像，不支持MRS V1.1作业管理接口，需要使用**V2作业管理接口**。

接口约束

无

调试

您可以在**API Explorer**中调试该接口。

URI

GET /v1.1/{project_id}/job-exes

表 8-4 路径参数

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

表 8-5 Query 参数

参数	是否必选	参数类型	描述
page_size	否	String	**参数解释：** 分页查询每页返回的最大作业数量。 **约束限制：** 不涉及 **取值范围：** 1～100 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
current_page	否	String	**参数解释：** 当前查询页码。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_name	否	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_id	是	String	**参数解释：** 集群编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	是否必选	参数类型	描述
state	否	String	**参数解释：** 作业状态编码。 **约束限制：** 不涉及 **取值范围：** ● -1：Terminated表示已终止的作业状态。 ● 2：Running表示运行中的作业状态。 ● 3：Completed表示已完成的作业状态。 ● 4：Abnormal表示异常的作业状态。 **默认取值：** 不涉及
id	否	String	**参数解释：** 作业执行对象的编号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求参数

无

响应参数

状态码： 200

表 8-6 响应 Body 参数

参数	参数类型	描述
totalRecord	Integer	**参数解释：** 作业列表总数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_executions	Array of **JobExeResult** objects	**参数解释：** 作业列表。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 8-7 JobExeResult

参数	参数类型	描述
id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_at	Long	**参数解释：** 作业创建时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
update_at	Long	**参数解释：** 作业更新时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenant_id	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_name	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
start_time	Long	**参数解释：** 作业执行开始时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	Long	**参数解释：** 作业执行结束时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
cluster_id	String	**参数解释：** 作业所属集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
group_id	String	**参数解释：** 作业执行组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
jar_path	String	**参数解释：** 执行程序jar包或sql文件地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
input	String	**参数解释：** 数据输入地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
output	String	**参数解释：** 数据输出地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_log	String	**参数解释：** 作业日志存储地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_type	Integer	**参数解释：** 作业类型码。 **约束限制：** 不涉及 **取值范围：** ● 1：MapReduce ● 2：Spark ● 3：Hive Script ● 4：HiveSQL（当前不支持） ● 5：DistCp ● 6：Spark Script ● 7：Spark SQL（该接口当前不支持） **默认取值：** 不涉及
file_action	String	**参数解释：** 导入导出数据。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
arguments	String	**参数解释：** 程序执行的关键参数。 **约束限制：** 该参数由用户程序内的函数指定，MRS只负责参数的传入。 **取值范围：** 该参数可为空。 **默认取值：** 不涉及

参数	参数类型	描述
hql	String	**参数解释：** HQL脚本语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_state	Integer	**参数解释：** 作业状态编码。 **约束限制：** 不涉及 **取值范围：** ● -1：Terminated表示已终止的作业状态。 ● 2：Running表示运行中的作业状态。 ● 3：Completed表示已完成的作业状态。 ● 4：Abnormal表示异常的作业状态。 **默认取值：** 不涉及
job_final_status	Integer	**参数解释：** 作业最终状态码。 ● 0：未完成。 ● 1：执行错误，终止执行。 ● 2：执行完成并且成功。 ● 3：已取消。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
hive_script_path	String	**参数解释：** Hive脚本地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_by	String	**参数解释：** 创建作业的用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
finished_step	Integer	**参数解释：** 当前已完成的步骤数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_main_id	String	**参数解释：** 作业主ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_step_id	String	**参数解释：** 作业步骤ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
postpone_at	Long	**参数解释：** 延迟时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
step_name	String	**参数解释：** 作业步骤名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
step_num	Integer	**参数解释：** 步骤数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
task_num	Integer	**参数解释：** 任务数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
update_by	String	**参数解释：** 更新作业的用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
spend_time	Float	**参数解释：** 作业执行持续时间，单位：秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
step_seq	Integer	**参数解释：** 步骤序列号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
progress	String	**参数解释：** 作业执行进度。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询作业exe对象列表请求示例

```
GET https://{endpoint}/v1.1/{project_id}/job-exes?
page_size=10&current_page=1&state=3&job_name=myfirstjob&clusterId=20ca8601-72a2-4570-
b788-2a20fec81a95
```

响应示例

状态码： 200

查询作业exe对象列表成功。

```
{
  "totalRecord" : "14",
  "job_executions" : [ {
    "id" : "669476bd-89d2-45aa-8f1a-872d16de377e",
    "create_at" : "1484641003707",
    "update_at" : "1484641003707",
    "tenant_id" : "3f99e3319a8943ceb15c584f3325d064",
    "job_id" : "",
    "job_name" : "myfirstjob",
    "start_time" : "1484641003707",
    "end_time" : null,
    "cluster_id" : "2b460e01-3351-4170-b0a7-57b9dd5ffef3",
    "group_id" : "669476bd-89d2-45aa-8f1a-872d16de377e",
    "jar_path" : "s3a://jp-test1/program/hadoop-mapreduce-examples-2.4.1.jar",
    "input" : "s3a://jp-test1/input/",
    "output" : "s3a://jp-test1/output/",
    "job_log" : "s3a://jp-test1/joblogs/",
    "job_type" : "1",
    "file_action" : "",
    "arguments" : "wordcount",
    "hql" : "",
    "job_state" : "2",
    "job_final_status" : "1",
    "hive_script_path" : null,
    "create_by" : "3f99e3319a8943ceb15c584f3325d064",
    "finished_step" : "0",
    "job_main_id" : "",
    "job_step_id" : "",
    "postpone_at" : "1484641003174",
    "step_name" : "",
    "step_num" : "0",
    "task_num" : "0",
    "update_by" : "3f99e3319a8943ceb15c584f3325d064",
    "spend_time" : null,
```

```
    "step_seq" : "222",
    "progress" : "first progress"
  } ]
}
```

状态码

状态码	描述
200	查询作业exe对象列表成功。

错误码

请参见**错误码**。

8.1.3 查询作业 exe 对象详情（废弃）

功能介绍

查询指定作业的exe对象详细信息。该接口不兼容Sahara。

说明

MRS 3.x版本镜像，不支持MRS V1.1作业管理接口，需要使用**V2作业管理接口**。

接口约束

无

调试

您可以在**API Explorer**中调试该接口。

URI

GET /v1.1/{project_id}/job-exes/{job_exe_id}

表 8-8 路径参数

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

请求参数

无

响应参数

状态码： 200

表 8-9 响应 Body 参数

参数	参数类型	描述
job_execution	**JobExeResult** object	**参数解释：** 作业详细信息。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

表 8-10 JobExeResult

参数	参数类型	描述
id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_at	Long	**参数解释：** 作业创建时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
update_at	Long	**参数解释：** 作业更新时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
tenant_id	String	**参数解释：** 项目编号。获取方法，请参见**获取项目ID**。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_id	String	**参数解释：** 作业ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_name	String	**参数解释：** 作业名称。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
start_time	Long	**参数解释：** 作业执行开始时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
end_time	Long	**参数解释：** 作业执行结束时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
cluster_id	String	**参数解释：** 作业所属集群ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
group_id	String	**参数解释：** 作业执行组ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
jar_path	String	**参数解释：** 执行程序jar包或sql文件地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
input	String	**参数解释：** 数据输入地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
output	String	**参数解释：** 数据输出地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_log	String	**参数解释：** 作业日志存储地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_type	Integer	**参数解释：** 作业类型码。 **约束限制：** 不涉及 **取值范围：** ● 1：MapReduce ● 2：Spark ● 3：Hive Script ● 4：HiveSQL（当前不支持） ● 5：DistCp ● 6：Spark Script ● 7：Spark SQL（该接口当前不支持） **默认取值：** 不涉及

参数	参数类型	描述
file_action	String	**参数解释：** 导入导出数据。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
arguments	String	**参数解释：** 程序执行的关键参数。 **约束限制：** 该参数由用户程序内的函数指定，MRS只负责参数的传入。 **取值范围：** 该参数可为空。 **默认取值：** 不涉及
hql	String	**参数解释：** HQL脚本语句。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_state	Integer	**参数解释：** 作业状态编码。 **约束限制：** 不涉及 **取值范围：** ● -1：Terminated表示已终止的作业状态。 ● 2：Running表示运行中的作业状态。 ● 3：Completed表示已完成的作业状态。 ● 4：Abnormal表示异常的作业状态。 **默认取值：** 不涉及

参数	参数类型	描述
job_final_status	Integer	**参数解释：** 作业最终状态码。 **约束限制：** 不涉及 **取值范围：** ● 0：未完成。 ● 1：执行错误，终止执行。 ● 2：执行完成并且成功。 ● 3：已取消。 **默认取值：** 不涉及
hive_script_path	String	**参数解释：** Hive脚本地址。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
create_by	String	**参数解释：** 创建作业的用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
finished_step	Integer	**参数解释：** 当前已完成的步骤数。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
job_main_id	String	**参数解释：** 作业主ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
job_step_id	String	**参数解释：** 作业步骤ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
postpone_at	Long	**参数解释：** 延迟时间，十三位时间戳。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
step_name	String	**参数解释：** 作业步骤名。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
step_num	Integer	**参数解释：** 步骤数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
task_num	Integer	**参数解释：** 任务数量。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
update_by	String	**参数解释：** 更新作业的用户ID。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
spend_time	Float	**参数解释：** 作业执行持续时间，单位：秒。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

参数	参数类型	描述
step_seq	Integer	**参数解释：** 步骤序列号。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及
progress	String	**参数解释：** 作业执行进度。 **约束限制：** 不涉及 **取值范围：** 不涉及 **默认取值：** 不涉及

请求示例

查询作业exe对象详情请求示例

```
GET https://{endpoint}/v1.1/{project_id}/job-exes/{job_exe_id}
```

响应示例

状态码： 200

查询作业exe对象详情成功。

```
{
  "job_execution" : {
    "id" : "632863d5-15d4-4691-9dc1-1a72340cb097",
    "create_at" : "1484240559176",
    "update_at" : "1484240559176",
    "tenant_id" : "3f99e3319a8943ceb15c584f3325d064",
    "job_id" : "632863d5-15d4-4691-9dc1-1a72340cb097",
    "job_name" : "hive_script",
    "start_time" : "1484240559176",
    "end_time" : null,
    "cluster_id" : "8b1d55f6-150e-45e2-8347-b2ca608d366b",
    "group_id" : "632863d5-15d4-4691-9dc1-1a72340cb097",
    "jar_path" : "s3a://jp-test1/program/Hivescript.sql",
    "input" : "s3a://jp-test1/input/",
    "output" : "s3a://jp-test1/output/",
    "job_log" : "s3a://jp-test1/joblogs/",
    "job_type" : "3",
    "file_action" : "",
    "arguments" : "wordcount",
    "hql" : null,
    "job_state" : "3",
    "job_final_status" : "1",
```

```
  "hive_script_path" : "s3a://jp-test1/program/Hivescript.sql",
  "create_by" : "3f99e3319a8943ceb15c584f3325d064",
  "finished_step" : "0",
  "job_main_id" : "",
  "job_step_id" : "",
  "postpone_at" : "1484240558705",
  "step_name" : "",
  "step_num" : "0",
  "task_num" : "0",
  "update_by" : "3f99e3319a8943ceb15c584f3325d064",
  "spend_time" : null,
  "step_seq" : "222",
  "progress" : "first progress"
 }
}
```


状态码

状态码	描述
200	查询作业exe对象详情成功。

错误码

请参见**错误码**。

8.1.4 删除作业执行对象（废弃）

功能介绍

删除指定的作业执行对象。该接口兼容Sahara。

说明

MRS 3.x版本镜像，不支持MRS V1.1作业管理接口，需要使用**V2作业管理接口**。

接口约束

无

调试

您可以在**API Explorer**中调试该接口。

URI

DELETE /v1.1/{project_id}/job-executions/{job_execution_id}

表 8-11 路径参数

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

请求参数

无

响应参数

无

请求示例

删除作业执行对象请求示例

```
DELETE https://{endpoint}/v1.1/{project_id}/job-executions/{job_execution_id}
```

响应示例

无

状态码

状态码	描述
204	删除作业执行对象成功。

错误码

请参见**错误码**。

9 权限策略和授权项

9.1 策略及授权项说明

如果您需要对您所拥有的MapReduce服务（MRS）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），如果华为账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用MRS服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为**角色**和**策略**。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询集群列表，那么这个IAM用户被授予的策略中必须包含允许“mrs:cluster: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，MRS集群管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。

- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。关于IAM项目与企业项目的区别，详情请参见：**IAM与企业管理的区别**。

  说明

  “√”表示支持，“x”表示暂不支持。

表 9-1 API 授权项列表

权限	对应API接口	授权项（Action）	IAM项目(Project)	企业项目(Enterprise Project)
创建集群并执行作业（V1）	POST /v1.1/{project_id}/run-job-flow	mrs:cluster:create	√	√
创建集群（V2）	POST/v2/{project_id}/clusters		√	√
查询集群列表（V1）	GET /v1.1/{project_id}/cluster_infos	mrs:cluster:list	√	√
获取集群列表V2接口（可获取集群详细信息）	GET/v2/{project_id}/clusters		√	√
删除集群	DELETE /v1.1/{project_id}/clusters/{cluster_id}	mrs:cluster:delete	√	√
查询主机列表（V1）	GET /v1.1/{project_id}/clusters/{cluster_id}/hosts	mrs:host:list	√	√
查询文件列表（V2）	GET/v2/{project_id}/clusters/{cluster_id}/files	mrs:file:list	√	√
新增作业并执行（V1）	POST /v1.1/{project_id}/jobs/submit-job	mrs:job:submit	√	√
新增并执行作业（V2）	POST /v2/{project_id}/clusters/{cluster_id}/job-executions		√	√
查询作业exe对象列表（V1）	GET /v1.1/{project_id}/job-exes	mrs:job:list	√	√

权限	对应API接口	授权项（Action）	IAM项目(Project)	企业项目(Enterprise Project)
查询单个作业信息（V2）	GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}		√	√
查询作业列表信息（V2）	GET /v2/{project_id}/clusters/{cluster_id}/job-executions		√	√
获取SQL结果（V2）	GET /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/sql-result		√	√
查询作业exe对象详情（V1）	GET /v1.1/{project_id}/job-exes/{job_exe_id}	mrs:job:get	√	√
查询用户代理信息	GET/v2/{project_id}/clusters/{cluster_id}/agency-mapping		√	√
查询作业日志详情	GET/v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/log-detail		√	√
查询指定集群的标签	GET /v1.1/{project_id}/clusters/{cluster_id}/tags	mrs:tag:list	√	√
查询所有标签	GET /v1.1/{project_id}/clusters/tags		√	√
创建单个集群标签	POST/v1.1/{project_id}/clusters/{cluster_id}/tags	mrs:tag:create	√	√
批量添加/删除集群标签	POST /v1.1/{project_id}/clusters/{cluster_id}/tags/action	mrs:tag:batchOperate	√	√
查询特定标签的集群列表	POST /v1.1/{project_id}/clusters/resource_instances/action	mrs:tag:listResource	√	×
终止作业（V2）	POST /v2/{project_id}/clusters/{cluster_id}/job-executions/{job_execution_id}/kill	mrs:job:stop	√	√
批量删除作业（V2）	POST /v2/{project_id}/clusters/{cluster_id}/job-executions/batch-delete	mrs:job:batchDelete	√	√
取消sql执行	POST/v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}/cancel	mrs:sql:cancel	√	√

权限	对应API接口	授权项（Action）	IAM项目(Project)	企业项目(Enterprise Project)
提交SQL语句	POST/v2/{project_id}/clusters/{cluster_id}/sql-execution	mrs:sql:execute	√	√
获取所有的弹性伸缩策略	GET/v2/{project_id}/autoscaling-policy/{cluster_id}	mrs:cluster:policy	√	√
配置弹性伸缩规则	POST /v1.1/{project_id}/autoscaling-policy/{cluster_id}		√	√
更新用户代理信息	PUT/v2/{project_id}/clusters/{cluster_id}/agency-mapping	mrs:cluster:syncUser	√	√
获取sql执行结果	GET/v2/{project_id}/clusters/{cluster_id}/sql-execution/{sql_id}	mrs:sql:get	√	√

相关链接

- MRS权限管理
- 创建用户并授权使用MRS
- MRS自定义策略

10 附录

10.1 MRS 所使用的弹性云服务器规格

针对不同的应用场景，MRS使用到如下类型的弹性云服务器。

- 通用计算增强型：C3型、C3ne型、C6型、C6s型、C7型
- 内存优化型：M3型、M6型、M7型
- 超高I/O型：I3型、IR3型
- 鲲鹏内存优化型：KM1型
- 鲲鹏超高I/O型：KI1型
- 鲲鹏通用计算增强型：KC1型

规格命名规则

AB.C.D

例如m2.8xlarge.8

其中，

- A表示系列，例如：s表示通用型、c表示计算型、m表示内存型。
- B表示系列号，例如：s1中的1表示通用型I代，s2中2表示通用型II代。
- C表示规格，当前系列中的规格大小，例如：medium、large、xlarge。
- D表示内存、CPU比，以具体数字表示，例如4表示内存和CPU的比值为4。

规格

表 10-1 通用计算增强型（C 型）弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
C3型	32	64	c3.8xlarge.2	KVM

类型	vCPU	内存(GB)	规格名称	虚拟化类型
C3型	16	64	c3.4xlarge.4	KVM
C3型	32	128	c3.8xlarge.4	KVM
C3型	60	256	c3ne.15xlarge.4	KVM
C3ne型	32	64	c3ne.8xlarge.2	KVM
C3ne型	16	64	c3ne.4xlarge.4	KVM
C3ne型	32	128	c3ne.8xlarge.4	KVM
C3ne型	60	256	c3ne.15xlarge.4	KVM
C6型	32	64	c6.8xlarge.2	KVM
C6型	64	128	c6.16xlarge.2	KVM
C6型	16	64	c6.4xlarge.4	KVM
C6型	32	128	c6.8xlarge.4	KVM
C6型	64	256	c6.16xlarge.4	KVM
C6s型	32	64	c6s.8xlarge.2	KVM
C6s型	64	128	c6s.16xlarge.2	KVM
C7型	32	64	c7.8xlarge.2	基于擎天架构的自研极简虚拟化
C7型	64	128	c7.16xlarge.2	基于擎天架构的自研极简虚拟化
C7型	128	256	c7.32xlarge.2	基于擎天架构的自研极简虚拟化
C7型	16	64	c7.4xlarge.4	基于擎天架构的自研极简虚拟化

类型	vCPU	内存(GB)	规格名称	虚拟化类型
C7型	32	128	c7.8xlarge.4	基于擎天架构的自研极简虚拟化
C7型	64	256	c7.16xlarge.4	基于擎天架构的自研极简虚拟化
C7型	128	512	c7.32xlarge.4	基于擎天架构的自研极简虚拟化

表 10-2 内存优化型弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
M3型	8	64	m3.2xlarge.8	KVM
M3型	16	128	m3.4xlarge.8	KVM
M3型	32	256	m3.8xlarge.8	KVM
M3型	60	512	m3.15xlarge.8	KVM
M6型	8	64	m6.2xlarge.8	KVM
M6型	16	128	m6.4xlarge.8	KVM
M6型	32	256	m6.8xlarge.8	KVM
M6型	64	512	m6.16xlarge.8	KVM
M7型	8	64	m7.2xlarge.8	基于擎天架构的自研极简虚拟化
M7型	16	128	m7.4xlarge.8	基于擎天架构的自研极简虚拟化

类型	vCPU	内存(GB)	规格名称	虚拟化类型
M7型	32	256	m7.8xlarge.8	基于擎天架构的自研极简虚拟化
M7型	64	512	m7.16xlarge.8	基于擎天架构的自研极简虚拟化
M7型	128	1024	m7.32xlarge.8	基于擎天架构的自研极简虚拟化

表 10-3 鲲鹏通用计算增强型（KC1 型）弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
KC1型	16	64	kc1.4xlarge.4	KVM
KC1型	32	64	kc1.8xlarge.2	KVM
KC1型	32	128	kc1.8xlarge.4	KVM
KC1型	60	120	kc1.15xlarge.2	KVM

表 10-4 鲲鹏内存优化型（KM1 型）弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
KM1型	8	64	km1.2xlarge.8	KVM
KM1型	16	128	km1.4xlarge.8	KVM
KM1型	32	256	km1.8xlarge.8	KVM
KM1型	60	480	km1.15xlarge.8	KVM

表 10-5 鲲鹏超高 I/O 型（KI1 型）弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
KI1型	16	64	ki1.4xlarge.4	KVM

类型	vCPU	内存(GB)	规格名称	虚拟化类型
KI1型	32	128	ki1.8xlarge.4	KVM
KI1型	64	228	ki1.16xlarge.4	KVM

表 10-6 超高 I/O 型弹性云服务器的规格

类型	vCPU	内存(GB)	规格名称	虚拟化类型
I3型	8	64	i3.2xlarge.8	KVM
I3型	16	128	i3.4xlarge.8	KVM
I3型	32	256	i3.8xlarge.8	KVM
I3型	64	512	i3.16xlarge.8	KVM
IR3型	16	64	ir3.4xlarge.4	KVM
IR3型	32	128	ir3.8xlarge.4	KVM

10.2 MRS 所使用的裸金属服务器规格

针对不同的应用场景，MRS使用到如下类型的裸金属服务器。

- 本地存储型（D2型）
- IO优化型（IO2）

规格命名规则

AB.C.D

例如m2.8xlarge.8

其中，

- A表示系列，例如：s表示通用型、c表示计算型、m表示内存型。
- B表示系列号，例如：s1中的1表示通用型I代，s2中2表示通用型II代。
- C表示规格，当前系列中的规格大小，例如：medium、large、xlarge。
- D表示内存、CPU比，以具体数字表示，例如4表示内存和CPU的比值为4。

规格

表 10-7 IO 优化型（IO2 型）裸金属服务器的规格

规格名称/ID	CPU	内存（GB）	本地磁盘	扩展配置
physical.io2.xlarge	2 * 22 Core Intel Xeon Gold 6161 V5 (2.2 GHz)	384 DDR4 RAM (GB)	2 * 800GB SSD RAID 1 + 10 * 800GB SSD	2 x 2 * 10GE

表 10-8 本地存储型（D2 型）裸金属服务器的规格

规格名称/ID	CPU	内存（GB）	本地磁盘	扩展配置
physical.d2.large	2 * 12 Core Intel Xeon Gold 5118 V5 (2.3 GHz)	192 DDR4 RAM (GB)	2 * 600GB SAS System Disk RAID 1 + 12 * 10TB SATA	2 x 2 * 10GE

10.3 状态码

状态码如表10-9所示。

表 10-9 状态码

状态码	编码	状态说明
100	Continue	继续请求。 这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。
101	Switching Protocols	切换协议。只能切换到更高级的协议。 例如，切换到HTTPS的新版本协议。
200	OK	服务器已成功处理了请求。
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。

状态码	编码	状态说明
204	NoContent	请求完全成功，同时HTTPS响应不包含响应体。 在响应OPTIONS方法的HTTPS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。
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HTTPS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。

状态码	编码	状态说明
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。
408	Request Time-out	服务器等候请求时发生超时。 客户端可以随时再次提交该请求而无需进行任何更改。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。
410	Gone	客户端请求的资源已经不存在。 返回该状态码，表明请求的资源已被永久删除。
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息。
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。

状态码	编码	状态说明
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	ServiceUnavailable	被请求的服务无效。 建议直接修改该请求，不要重试该请求。
504	ServerTimeout	请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。
505	HTTPS Version not supported	服务器不支持请求的HTTPS协议的版本，无法完成处理。

10.4 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见**API网关错误码**进行处理。

状态码	错误码	错误信息	描述	处理措施
400	0023	Failed to obtain cluster details.	获取集群详情信息失败!	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常，请检查绑定的EIP是否存在，使用正确的EIP进行绑定
400	0056	Jobs cannot be submitted to a cluster in the xxx state.	集群状态为xxx，不能提交作业!	请等待集群任务执行完成且状态变为运行中
400	0057	Spark jobs cannot be submitted.	不能提交Spark相关作业!	请检查是否存在Spark服务以及Spark服务是否正常
400	0093	Metadata of version xxx not found.	无法找到版本号为xxx的版本元数据信息。	请确认集群版本是否正确并使用正确的集群版本

状态码	错误码	错误信息	描述	处理措施
400	0160	Failed to kill the job.	终止作业失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0161	Failed to delete jobs in batches.	批量删除作业失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0162	Failed to query the job.	查询作业失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0165	Failed to verify the SQL statement.	校验SQL语句失败	检查sql语句是否正确
400	0166	Failed to query a job list.	查询作业列表失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0167	The v2 job API cannot be accessed.	不支持访问v2作业接口	在该集群版本中请使用v1接口
400	0168	Hive jobs cannot be submitted.	不能提交Hive相关作业	请检查是否存在Hive服务以及Hive服务是否正常
400	0169	Flink jobs cannot be submitted.	不能提交Flink作业	请检查是否存在Flink服务以及Flink服务是否正常
400	0170	Failed to collect job log directory information.	收集作业日志目录信息失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0171	Failed to collect job log details.	收集作业日志详情失败	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	0172	Failed to collect the SQL job result.	收集SQL作业结果失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常

状态码	错误码	错误信息	描述	处理措施
400	0173	Failed to submit the job (cluster ID: xxx; job name: xxx).	执行提交作业操作失败，集群ID为xxx，作业名字为xxx	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0174	Failed to query the job.	查询作业失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0175	Failed to kill the job.	终止作业失败	请尝试重新发起调用，或联系技术支持
400	0176	The job does not exist.	作业不存在	按照报错提示修改请求参数
400	0177	The number of jobs running in each cluster cannot exceed 10.	每个集群运行作业的数量不能超过10	按照报错提示修改请求参数
400	0178	The job ID cannot be left blank.	作业Id不可为空	按照报错提示修改请求参数
400	0179	The job type must be SparkSql or SparkScript.	作业类型必须是SparkSql或者SparkScript	按照报错提示修改请求参数
400	0180	The job is being submitted.	作业正在提交中	请等待完成后再进行相关操作
400	0181	The SQL job result collection is empty.	SQL作业结果收集为空	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0182	Failed to read the SQL job result.	读取SQL作业结果失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0183	The job is running.	作业正在运行中	请等待完成后再进行相关操作
400	0185	The log type does not exist.	日志类型不存在	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	0187	The log aggregation path is empty. Logs cannot be queried.	作业日志聚集路径为空，无法查询日志	按照检查作业日志路径是否正确
400	0188	The job query result is empty. Logs cannot be obtained.	作业查询结果为空，无法获取日志	按照检查作业日志路径是否正确
400	0189	Failed to delete the job list.	删除作业列表失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	0190	The user who submits the job cannot be empty.	提交作业用户不能为空	按照报错提示修改请求参数
400	0191	Failed to query the user who submits the job on MRS Manager.	在MRS Manager查询提交作业用户失败	按照报错提示修改请求参数
400	0192	The current user does not exist on MRS Manager. Grant the user sufficient permissions on IAM and then perform IAM user synchronization on the Dashboard tab page.	当前用户在MRS Manager不存在。请先在IAM给予该用户足够的权限，再在概览页签进行IAM用户同步。	按照报错提示进行处理
400	0193	Failed to operate the database job records.	操作数据库作业记录失败	请尝试重新发起调用，或联系技术支持
400	0194	Failed to start the launcher and submit the job.	作业启动失败，作业提交不成功	请尝试重新发起调用，或联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	0199	Failed to delete the jobs.	删除作业失败	请尝试重新发起调用，或联系技术支持
400	0200	Failed to kill the jobs on Yarn.	Yarn上终止作业失败	请尝试重新发起调用，或联系技术支持
400	0201	The job does not exist.	作业不存在	按照报错提示修改请求参数
400	0202	Too many jobs are being submitted, please try again later.	当前提交作业过多，请稍后重试	请尝试重新发起调用，或联系技术支持
400	0211	The maximum number of bound security groups has been reached. A maximum of four security groups can be bound, excluding the default security group that is automatically created.	绑定安全组数量超出限制，最多允许绑定4个，不含自动创建的默认安全组	按照报错提示修改请求参数
400	12000002	The parameter is invalid.	参数不合法。	请对照API文档检查参数是否正确
400	12000003	The cluster does not exist.	集群不存在!	请检查集群是否存在
400	12000009	The parameter is invalid.	参数不合法!	请对照API文档检查参数是否正确

状态码	错误码	错误信息	描述	处理措施
400	12000012	Tasks are being executed in the cluster. Subcontract period cluster is not allowed for clusters that are in this state. Scale-out or scale-in is not allowed for clusters that are not in the running state.	集群已有任务执行中！ 当前状态不允许转包周期集群。状态不是【运行中】的集群，不允许执行缩容/扩容操作！	请等待集群任务执行完成且状态变为运行中
400	12000013	The task node does not exist.	task节点不存在！	请检查对应节点是否存在
400	12000014	Failed to scale out the cluster (ID: xxx). The type and the quantity of nodes to be added are xxx and xxx, respectively.	执行扩容集群操作失败，集群ID为xxx，扩容节点类型为xxx，扩容数量为xxx。	请尝试重新发起调用，或联系技术支持。
400	12000018	Scale-out or scale-in cannot be performed again because it is in progress.	缩容/扩容中的集群，不允许再次缩容/扩容！	请等待集群任务执行完成且状态变为运行中
400	12000019	Failed to obtain hosts of the cluster.	获取集群主机列表失败！	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000020	Failed to terminate the cluster.	终止集群失败！	请尝试重新发起调用，或联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	12000021	Clusters in the xxx state cannot be terminated.	不允许终止状态为xxx的集群！	请等待集群任务执行完成且状态变为运行中
400	120000212	Failed to obtain the AZs that you have permission to access.	获取有权限可用区失败。	请尝试重新发起调用，或联系技术支持
400	120000213	The region [xxx] does not exist.	指定的Region[xxx]不存在。	按照报错提示修改请求参数
400	120000214	No permission to access the AZs [xxx]. The following AZs [xxx] can be accessed.	无权限访问如下可用区[xxx]，可访问可用区[xxx]。	按照报错提示修改请求参数
400	12000023	Failed to obtain cluster details.	获取集群详情信息失败！	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000024	Yearly/ Monthly clusters cannot be terminated.	包年包月的集群不可终止！	请避免进行此操作
400	12000027	Failed to verify the subnet when creating the cluster xxx.	创建集群xxx，子网校验失败！	请检查子网可用数量是否满足创建集群所需要求
400	12000028	The cluster has a maximum of xxx Core and Task nodes.	集群Core和Task节点总数最多为xxx个	尝试减少请求中所要求的节点数量
400	12000029	Failed to obtain the quota.	获取配额失败	请尝试重新发起调用，或联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	12000030	The requested number of nodes in the cluster exceeds the available quota.	申请集群的节点总数已大于可用配额数！	尝试减少请求中所要求的资源量，或者增加可用配额
400	12000031	The requested number of vCPUs in the cluster exceeds the available quota.	申请集群的CPU总核数已大于可用配额数！	尝试减少请求中所要求的资源量，或者增加可用配额
400	12000032	The requested memory of the cluster exceeds the available quota.	申请集群的内存总数已大于可用配额数！	尝试减少请求中所要求的资源量，或者增加可用配额
400	12000033	The requested number of disks in the cluster exceeds the available quota.	申请集群的磁盘块数已大于可用配额数！	尝试减少请求中所要求的资源量，或者增加可用配额
400	12000034	The requested disk capacity of the cluster exceeds the available quota.	申请集群的磁盘容量已大于可用配额数！	尝试减少请求中所要求的资源量，或者增加可用配额
400	12000036	Failed to obtain product information.	获取产品信息失败！	请尝试重新发起调用，或联系技术支持
400	12000038	Failed to obtain the security group.	获取安全组失败！	请尝试重新发起调用，或联系技术支持
400	12000041	Failed to obtain the cluster list.	获取集群列表信息失败！	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常

状态码	错误码	错误信息	描述	处理措施
400	12000042	Failed to create a cluster.	创建集群失败！	请尝试重新发起调用，或联系技术支持，或请对照API文档检查参数是否正确
400	12000043	Duplicate cluster name: xxx.	存在相同的集群名称：xxx。	请使用其他集群名称
400	12000044	The minimum memory of a Master node is xxx GB.	集群的Master节点内存数最小为xxxGB。	请增加Master节点内存大小
400	12000045	Insufficient quota of the security group.	安全组配额不足！	请增加安全组配额
400	12000046	Insufficient quota of the security group rule.	安全组规则配额不足！	请增加安全组规则配额，或者删除不必要的安全组规则
400	12000048	Product specification xxx does not exist.	产品规格 xxx 不存在！	请更换其他产品规格或可用区
400	12000050	Incorrect certificate.	错误的证书！	请更换正确的证书
400	12000052	No access rights.	无权访问！	请检查权限是否满足要求
400	12000053	Invalid billing type.	无效的订单类别！	请对照API文档检查参数是否正确
400	12000054	The operation is not supported.	不支持此操作！	请避免进行此操作
400	12000055	Failed to open the file.	打开文件失败！	请尝试重新发起调用，或联系技术支持
400	12000059	Key pair xxx does not exist.	用户密钥对xxx不存在。	请检查密钥对是否存在，更换正确的密钥对

状态码	错误码	错误信息	描述	处理措施
400	12000060	The number of running jobs per cluster cannot exceed xxx.	每个集群运行作业的数量不能超过xxx!	按照报错提示修改请求参数
400	12000061	Failed to submit the job (cluster ID: xxx; job name: xxx; job ID: xxx).	执行提交作业操作失败，集群ID为xxx，作业名字为xxx，ID为xxx。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000062	Jobs cannot be submitted to a cluster in the xxx state.	集群状态为xxx，不能提交作业!	请等待集群任务执行完成且状态变为运行中
400	12000063	Spark jobs cannot be submitted.	不能提交Spark相关作业!	请检查是否存在Spark服务以及Spark服务是否正常
400	12000064	Jobs cannot be submitted or inquired to a security cluster by API.	安全集群不能调用API提交/查询作业!	请使用非安全集群或者使用客户端等其他方式提交作业
400	12000065	Scale-out or scale-in is not available for Yearly/ Monthly clusters.	包年包月的集群不支持扩容/缩容!	请避免进行此操作
400	12000068	If the job type is Hive or Spark Script, the value of the mains should not be left blank.	作业类型为Hive或者Spark Script时,mains不能为空!	按照报错提示修改请求参数
400	12000069	If the job type is MapReduce or Spark, the value of the libs should not be left blank.	作业类型为MapReduce或者Spark时,libs不能为空!	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	12000070	An error occurred while accessing Knox.	访问knox发生异常。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000071	The Executor server has an internal error.	Executor服务器内部错误。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000073	Failed to access the HDFS directory.	HDFS目录连接失败!	请尝试重新发起调用，或联系技术支持，或检查集群HDFS服务是否正常
400	12000077	The password of [xxx] must meet the following requirements: 1.must be xxx to xxx characters long. 2.at least contain [xxx] types of the following characters: uppercase letters, lowercase letters, digits, and special characters(`~!@#$%%^&*()-_=+|[{}];:',<.>?). 3.cannot be the username or the username spelled backwards.	[xxx]密码必须符合以下规则：1.字符长度为[xxx-xxx]. 2.至少包含以下字符中的[xxx]种：大写字母、小写字母、数字和特殊字符(`~!@#$%%^&*()-_=+|[{}];:',<.>?). 3.不能与用户名或倒序的用户名相同。	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	12000080	The status of some nodes is not running in the cluster. Try again later.	集群存在非运行状态节点，请稍后重试。	请尝试重新发起调用，或联系技术支持
400	12000081	The jar_path parameter cannot be left blank if the job type is MapReduce or Spark.	当作业类型为MapReduce或Spark时，jar_path参数不能为空。	请补充jar_path参数
400	12000082	Node groups cannot be deleted in the cluster that is being scaled in or scaled out.	缩容/扩容中的集群，不允许删除节点组。	请等待集群任务执行完成且状态变为运行中
400	12000085	This interface does not support the cluster of this version.Please use the /v2/{project_id}/clusters/{cluster_id}/job-executions job submit interface.	该接口不支持该版本的集群，请使用/v2/{project_id}/clusters/{cluster_id}/job-executions作业提交接口。	请使用符合该集群版本的新接口
400	12000086	This cluster version not support s3[an]: schema. Please use the obs: schema.	此集群版本不支持s3[an]:协议，请使用obs:协议。	请使用符合该集群版本的协议
400	12000087	Failed to get billing records.	获取话单文件失败。	请尝试重新发起调用，或联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	12000090	Products of the xxx specifications are no longer available in selected AZ.	在选中的可用区域，产品规格为xxx的产品已下架。	请更换其他产品规格或可用区
400	12000092	Failed to get metadata of version xxx.	获取版本号为xxx的版本元数据失败。	请尝试重新发起调用，或联系技术支持
400	12000093	Metadata of version xxx not found.	无法找到版本号为xxx的版本元数据信息。	请确认集群版本是否正确并使用正确的集群版本
400	12000094	The xxx in xxx version does not support the xxx flavor.	xxx节点在xxx版本不支持xxx规格。	请更换其他产品规格
400	12000095	Patch xxx is unavailable.	补丁xxx不可用。	请检查补丁版本号是否正确
400	12000099	Topology template of version xxx not found.	无法找到版本号为xxx的拓扑模板信息。	请确认集群版本是否正确并使用正确的集群版本
400	12000100	Failed to stop the cluster. Only running or abnormal clusters can be stopped.	停止集群失败，只允许停止“运行中”或“异常”状态的集群。	请等待集群任务执行完成且状态变为'运行中'或'异常'
400	12000101	Failed to start the cluster. Only stopped clusters can be started.	启动集群失败，只允许启动“已停止”状态的集群。	请等待集群任务执行完成且状态变为'已停止'
400	12000102	Failed to stop the cluster.	停止集群失败。	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	12000103	Failed to start the cluster.	启动集群失败。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常

状态码	错误码	错误信息	描述	处理措施
400	12000104	VPC|ECS|DSS|DCC|OpenStack service error.	VPC|ECS|DSS|DCC|Openstack服务错误。	请尝试重新发起调用，或联系技术支持
400	12000105	VPC xxx does not exist.	虚拟私有云xxx不存在。	请检查虚拟私有云是否存在
400	12000106	Key pair xxx does not exist.	用户密钥对xxx不存在。	请检查密钥对是否存在，更换正确的密钥对
400	12000107	Invalid project ID: xxx.	非法的project id: xxx。	请使用正确的projectId
400	12000108	Failed to verify the EIP when creating the cluster xxx.	创建集群xxx，EIP校验失败!	请检查绑定的EIP是否存在，使用正确的EIP进行绑定
400	12000109	Failed to bind the EIP to cluster xxx.	集群xxx绑定弹性公网IP失败。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常，请检查绑定的EIP是否存在，使用正确的EIP进行绑定
400	12000110	Failed to unbind the EIP from cluster xxx.	集群xxx解绑弹性公网IP失败。	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
400	12000111	Failed to bind the EIP. The EIP xxx has been bound to another resource.	弹性公网IP绑定失败，此弹性公网IPxxx已经绑定到其他资源。	请使用其他未被占用的弹性公网IP
400	12000112	The EIP xxx does not exist.	弹性公网IPxxx不存在。	请检查绑定的EIP是否存在，使用正确的EIP进行绑定
400	12000113	Failed to update the EIP.	弹性公网IP更新失败。	请检查绑定的EIP是否存在，使用正确的EIP进行绑定

状态码	错误码	错误信息	描述	处理措施
400	12000114	The hive_script_path parameter cannot be left blank if the job type is Hive or SparkScript.	当作业类型为Hive或SparkScript时，hive_script_path参数不能为空。	请按照提示补充缺失的参数
400	12000115	The ECS group quota is insufficient.	云服务器组配额不足。	请增加相关资源的配额
400	12000116	The VPC quota is insufficient. Select the existing VPC or increase the quota.	VPC配额不足，请选择现有vpc或申请更多配额。	请增加相关资源的配额
400	12000117	The subnet quota is insufficient. Select the existing subnet or increase the quota.	子网配额不足，请选择现有子网或申请更多配额。	请增加相关资源的配额
400	12000118	Failed to create the security group rule.	添加安全组规则失败！	请检查安全组规则是否已满，或稍后重试
400	12000119	The security group rule already exists.	安全组规则已存在。	无需进行此操作或者手动修改安全组

状态码	错误码	错误信息	描述	处理措施
400	12000121	Failed to submit a request to scale out the Yearly/Monthly cluster. Scale-out is not allowed because the cluster has an unpaid order. Scale out the cluster again after you pay the order.	扩容包周期集群请求提交失败。 该集群存在未支付订单，不允许扩容，请完成订单后再执行操作。	请完成订单后再执行操作
400	12000122	EPS service error.	EPS服务错误。	请尝试重新发起调用，或联系技术支持
400	12000123	Failed to update Task node information because the number of Task nodes is not 0.	由于task节点数量不为0，无法更新task节点信息。	请缩容节点组使节点组里的个数为0后，再进行相关操作
400	12000124	In the cluster xxx, the number of Task nodes can be adjusted only using auto scaling.	集群xxx只支持使用弹性伸缩来调整task节点数量	按照报错提示进行处理
400	12000125	Failed to update Task node information, because the cluster state is scaling out or scaling in.	由于集群状态为扩容或者缩容的状态，无法更新task节点信息。	请等待集群任务执行完成且状态变为运行中
400	12000126	Failed to obtain authentication information.	获取鉴权信息失败。	请尝试重新发起调用，或联系技术支持

状态码	错误码	错误信息	描述	处理措施
400	12000127	Failed to lock cluster operation.	限制集群操作失败。	请尝试重新发起调用，或联系技术支持
400	12000128	Failed to unlock cluster operation.	解除限制集群操作失败。	请尝试重新发起调用，或联系技术支持
400	12000129	Master node specifications cannot be scaled up for a cluster that is not in the running state.	状态不是【运行中】的集群，不允许执行Master节点升级规格操作!	请等待集群任务执行完成且状态变为运行中
400	12000130	Specifications available for scale-up not found.	找不到可以升级的节点规格。	请更换其他产品规格
400	12000131	Master node specifications cannot be scaled up for a non-HA cluster.	不支持升级非高可用的集群的Master节点。	按照报错提示进行处理
400	12000132	vCPUs and memory cannot be reduced in the specification scale-up.	升级节点规格不支持减小vcpu数/memory数。	按照报错提示进行处理
400	12000133	Specification scale-up is not available for this type of nodes.	升级节点规格暂时不支持此节点类型。	按照报错提示进行处理
400	12000134	Failed to scale up the Master node specifications.	升级Master节点规格失败。	请尝试重新发起调用，或联系技术支持
400	12000135	Master nodes available for specification scale-up not found	找不到可以升级的Master节点	请检查对应节点是否存在

状态码	错误码	错误信息	描述	处理措施
400	12000138	Failed to get max server Group members.	查询ECS反亲和组最大虚机数失败。	请尝试重新发起调用，或联系技术支持
400	12000139	All evs volume type is sellout, please try again later.	当前所有的云硬盘已经售罄, 请稍后重试。	请尝试重新发起调用，或联系技术支持
400	12000140	Evs volume type:xxx is sellout, please try again later.	当前云硬盘:xxx已经售罄, 请稍后重试。	请尝试重新发起调用，或联系技术支持
400	12000141	The disk size of a node cannot be less than {value} GB.	节点磁盘大小不能小于{value}GB！	按照报错提示进行处理
400	12000142	The disk size cannot exceed 32,000 GB.	磁盘大小不能大于32000GB！	按照报错提示进行处理
400	12000154	IAM synchronization is in progress and cannot be triggered again in the same cluster. Cluster ID: xxx	IAM同步正在进行中，同一集群不能再次触发，集群ID为xxx。	请等待完成后再进行相关操作
400	12000156	Failed to query Iam group	查询IAM用户组失败	请尝试重新发起调用，或联系技术支持
400	12000157	Failed to query Iam user or role	查询IAM角色和用户失败	请尝试重新发起调用，或联系技术支持
400	12000163	Failed to query Manager user.	查询Manager用户失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常

状态码	错误码	错误信息	描述	处理措施
400	12000164	Failed to query Manager user group.	查询Manager用户组失败	请尝试重新发起调用，或联系技术支持，或检查集群状态是否正常
400	12000209	Either a VPC ID or name is required.	虚拟私有云ID和名字至少需要一个。	按照报错提示进行处理
400	12000210	Either a subnet ID or name is required.	子网ID和名字至少需要一个。	按照报错提示进行处理
400	12000233	Insufficient resources for flavor xxx. Reduce the purchase quantity and try again. Alternatively, select another instance type or flavor, or switch the AZ or region to select your desired product.	申请集群的规格：xxx资源不足，请减少购买量后重试，或选择其他类型与规格的实例，或切换区域、可用区重新选购！	请减少购买量后重试，或选择其他类型与规格的实例，或切换区域、可用区重新选购
400	12000234	Insufficient resources for the flavor xxx of the node to be scaled out. Reduce the purchase quantity and try again.	扩容节点的规格：xxx资源不足，请减少购买量后重试！	请减少购买量后重试或更换其他规格
400	12000360	Cluster name cannot be updated for a cluster that is in the terminating or terminated state.	状态是【删除中/删除】的集群，不允许修改集群名字操作！	请等待集群任务执行完成且状态变为运行中

状态码	错误码	错误信息	描述	处理措施
400	12000361	Only offline cluster is allowed to modify manager access ip.	非纳管集群不允许修改Manager页面访问IP	按照报错提示进行处理
400	12000362	Cluster id is already exist or not universally unique identifier format.	集群ID已存在或者非UUID格式!	按照报错提示使用正确的请求参数
400	12000363	The offline cluster with same management node IPs[xxx] already exists.	已存在相同的管理节点IP[xxx]的纳管集群!	按照报错提示修改请求参数
400	12000405	The number of nodes where the xxx role is deployed must be greater than or equal to xxx and less than or equal to xxx. The step is xxx.	xxx角色部署的节点个数必须满足：大于等于xxx个节点，小于等于xxx个节点，步长为xxx。	按照报错提示修改请求参数
400	12000406	The number of nodes where the xxx role is deployed must be greater than or equal to xxx and less than or equal to xxx.	xxx角色部署的节点个数必须满足：大于等于xxx个节点，小于等于xxx个节点。	按照报错提示修改请求参数
400	12000407	The xxx role must be deployed on xxx nodes.	xxx角色部署的节点个数必须是：xxx个节点。	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	12003001	The service xxx was installed.	xxx服务已安装。	按照报错提示修改请求参数
400	12003002	The service xxx is not in version metadata.	元数据中没有xxx服务。	请尝试重新发起调用，或联系技术支持。
400	12003003	The group [xxx] is not exist	节点组xxx不存在!	请使用集群中存在的节点组
400	12003004	The cluster does not support add components.	添加服务失败。	请尝试重新发起调用，或联系技术支持。
400	12003005	The role xxx is not belong to the service xxx.	xxx角色不属于xxx服务。	按照报错提示修改请求参数
400	12003006	The cluster is not in the [running] state.	集群不是【运行中】!	请等待集群任务执行完成且状态变为运行中
400	12003008	Occur service xxx that depend on xxx, can not add or delete.	xxx依赖xxx，不能添加或删除。	按照报错提示修改请求参数
400	12003009	Can not add services for nodegroup xxx that doesn't have normal hosts.	不能给节点组xxx添加服务，该节点组没有状态正常的主机。	请尝试重新发起调用，或联系技术支持。
400	12003012	xxx can not deploy to abnormal host, please check.	不能在状态不正常的主机上部署xxx，请检查。	按照报错提示修改请求参数
400	12003013	Can not deploy xxx to xxx, the host state is not normal.	不能给主机xxx部署xxx，该主机的状态不正常。	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	12003021	Can not add the stateful roles [xxx] to the group whose type is task	如下有状态的角色【xxx】不能添加到类型为Task的节点组中	请检查节点组类型或角色状态。
400	12003022	There are multiple roles [xxx] whose local_disks_anti_affinity is true.In strict anti-affinity mode, a node group can have only one role with local_disks_anti_affinity set to true.	当前存在多个local_disks_anti_affinity为true的角色【xxx】，严格模式反亲一个节点组只能有一个local_disks_anti_affinity为true的角色	按照报错提示修改请求参数
400	12003023	The following components [xxx] cannot be added to the security cluster where Kerberos authentication is enabled.	如下组件【xxx】不能添加到开启kerberos认证的安全集群中。	请检查组件或集群类型。
400	12003054	The xxx role is missing. Check the request parameters.	缺少xxx角色，请检查请求参数。	按照报错提示修改请求参数
400	12003090	The index must be greater than zero and an integer.	角色的索引必须大于零且为整数。	按照报错提示修改请求参数
400	12005001	The number of tags in one cluster must not exceed xxx.	单个集群中的tag个数不能大于xxx!	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	12005002	Tag xxx does not comply with the specifications. The tag key and value cannot start or end with a space and cannot contain any of the following characters: =*<>\\,|/.	标签xxx不符合规范，标签的key和value不能包含“=”，“*”，“<”，“>”，“\\”，“,”，“|”，“/”，且首尾字符不能为空格!	按照报错提示修改请求参数
400	130000002	The token is invalid.	Token无效	请更新最新Token或检查token是否正确
400	13000046	Security group xxx does not exist.	安全组 xxx 不存在!	请检查安全组是否存在
400	MRS.00005045	Role xxx must be deployed on the master node.	xxx角色必须部署在master节点。	按照报错提示修改请求参数
400	MRS.00005058	Current cluster version: xxx does not support config clickhouse password.	当前集群版本【xxx】不支持配置ClickHouse密码。	按照报错提示修改请求参数
400	MRS.00005059	Security Cluster does not support config clickhouse password.	安全集群不支持配置ClickHouse密码。	按照报错提示修改请求参数
400	MRS.00005060	Current Service: xxx does not support config password.	当前服务【xxx】不支持配置密码。	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	MRS.00005061	Clickhouse user password or default user password can not be empty.	ClickHouse用户或者default用户密码不能为空。	按照报错提示修改请求参数
400	MRS.00005094	Role xxx must be deployed on the non-master node.	xxx角色必须部署在非master节点。	按照报错提示修改请求参数
400	MRS.00005095	Role xxx must be deployed on all nodes.	xxx角色必须部署在所有节点上。	按照报错提示修改请求参数
400	MRS.00005096	The role metadata is abnormal.	角色元数据异常。	请尝试重新发起调用，或联系技术支持。
400	MRS.00005097	The number of xxx role instances must be greater than or equal to 0 and less than or equal to xxx.	xxx角色实例数必须大于等于0，小于等于xxx	按照报错提示修改请求参数

状态码	错误码	错误信息	描述	处理措施
400	MRS.0010	Dataconnector error.	连接器的名称不合法，名称只能包含字母大小写、数字、-_！ 连接器的ID不能为空！ 不支持的数据类型: xxx！ 不支持的组件数据类型: xxx! 没有找到连接ID为xxx的数据连接实例！ 没有找到ID为xxx的集群与数据连接映射！ 没有找到连接ID为xxx的集群与数据连接映射！ RDS的实例ID不能为空！ RDS的实例、数据库名、用户名和密码都不能为空！ OBS的AccessKey、SecretKey、目录都不能为空！ JDBC的IP、端口、数据库、用户名、密码都不能为空！ 没有找到xxx可用的驱动！ 数据连接的类型不匹配，传入的类型：xxx，实际数据连接的类型为xxx！ RDS实例虚拟私有云的ID xxx与集群的虚拟私有云ID xxx不一致！ RDS实例子网的ID xxx与集群的子网ID	请按照报错提示进行检查和处理

状态码	错误码	错误信息	描述	处理措施
			xxx不一致！ RDS实例的安全组ID xxx与集群的安全组ID xxx不一致！ 集群已经存在相同位置的数据连接，xxx! 数据连接已经被其他集群使用（xxx）！已经存在名称为xxx的数据连接! 集群无法连接此RDS实例的数据库，请检查网络/数据库名称/用户名/密码！ 没有找到ID为xxx的RDS实例！ RDS实例xxx的状态异常，状态xxx！ 同步数据连接参数到集群失败！	

状态码	错误码	错误信息	描述	处理措施
400	MRS.0011	SQL typed xxx can not run on the cluster which not installed dependent components. Cannot execute SQL on a cluster in the xxx state. Not support sql execution in cluster version xxx. Request with multiple SQL is not support. Failed to submit the SQL request to Executor (cluster ID: xxx).	SQL类型为xxx的语句无法在未安装相关依赖组件的集群中运行！ 集群状态为xxx，不能执行SQL！ 集群版本xxx不支持执行SQL接口！ 不支持同时提交多条SQL！ 提交SQL到Executor上失败，集群ID为xxx！	按照报错提示修改请求参数
400	MRS.0015	The scaling operation failed due to not meeting the scaling conditions.	不符合扩缩容条件导致扩缩容请求失败。	请根据返回体内容查看详细原因，或联系技术支持
400	MRS.0016	The request failed due to restrictions related to the cluster node group.	因集群节点组不符合特定条件导致请求失败	请根据返回体中的错误信息进行对应处理，或者联系技术支持
400	MRS.0020	The cluster does not support add components.	集群不支持添加组件功能。	请检查集群版本或者集群类型
400	MRS.0205	Failed to sync agency mapping configuration to cluster.	同步委托映射配置到集群失败！	请检查委托是否正确

状态码	错误码	错误信息	描述	处理措施
400	MRS.0206	Updating agency mapping task is running.	正在更新映射配置任务！	请等待完成后再进行相关操作
400	MRS.0207	Parse Json format failed.	解析JSON失败！	请检查json是否正确
400	MRS.0208	Create or modify policy failed.	创建/修改策略失败。	请检测策略是否正确
400	MRS.0209	Assign policy to agency failed.	给委托赋予策略失败。	请检查委托是否正确
400	MRS.0210	No secu_admin policy.	没有secu_admin权限。	请按照报错提示补充相关权限
400	MRS.0211	Failed to obtain new agency or new agency does not exist.	新委托获取失败/不存在。	请检查委托是否正确
400	MRS.0212	Updating ECS agency task is running.	正在更新ECS委托！	请尝试重新发起调用，或联系技术支持
400	MRS.0216	Failed to update ECS metadata.	ecs元数据更新失败。	请尝试重新发起调用，或联系技术支持
400	MRS.0217	Failed to unbind policy.	策略解绑失败。	请检查策略是否正确
400	MRS.0218	Failed to check whether policies are exist or not.	校验策略存在失败。	请检查策略是否正确
400	MRS.1010	The RDS instance whose ID is xxx is not found. The status (xxx) of RDS instance xxx is abnormal.	没有找到ID为xxx的RDS实例！ RDS实例xxx的状态异常，状态xxx！	按照报错提示进行处理

状态码	错误码	错误信息	描述	处理措施
401	12000001	Invalid authentication.	认证失败	请检查账户状态和认证内容是否正常
401	12000136	Permission denied. Error message: Policy doesn't allow bss:order:update to be performed.	您的权限不足。Error message: Policy doesn't allow bss:order:update to be performed.	请按照报错提示补充相关权限
403	12000057	Failed to obtain the file list.	获取文件列表失败!	请检查MRS集群状态、Master节点状态、集群网络通信、安全组规则是否正常
404	12005003	The tag key xxx does not exist in cluster xxx.	标签的键xxx在集群xxx中不存在!	按照报错提示修改请求参数
500	12000004	Internal server error.	服务器内部错误	请尝试重新发起调用，或联系技术支持

10.5 获取项目 ID

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（ project_id ），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“我的凭证”。
   在“API凭证”页面的项目列表中查看项目ID。

多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。

图 10-1 查看项目 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，具体请参考**终端节点**。

响应示例如下，其中projects下的“id”即为“name”所对应区域的项目ID。

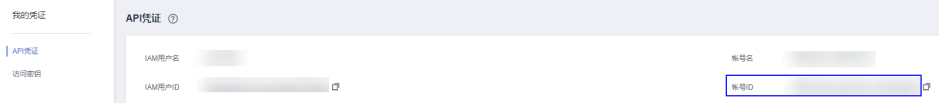
```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "region_id",
            "description": "",
            "links": {
                "next": null,
                "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"
    }
}
```

10.6 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID（domain-id），所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。
   在“API凭证”页面中查看账号ID。

图 10-2 获取账号 ID



10.7 获取 MRS 集群信息

MRS 服务支持的组件

- MRS 3.2.0-LTS.1支持的组件信息如下：
  - 分析集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez,Guardian
  - 流式集群包含的组件有：Kafka,Flume,ZooKeeper,Ranger
  - 混合集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez,Kafka,Flume,Guardian
  - 自定义集群包含的组件有：CDL,Hadoop,Spark2x,HBase,Hive,Hue,IoTDB,Loader,Kafka,Flume,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez,ClickHouse,Guardian
- MRS 3.1.5支持的组件信息如下：
  - 分析集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,kudu,Sqoop,Guardian
  - 流式集群包含的组件有：Kafka,Flume,ZooKeeper,Ranger
  - 混合集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,kudu,Sqoop,Guardian,Kafka,Flume
  - 自定义集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Kafka,Flume,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,ClickHouse,kudu,Sqoop,Guardian
- MRS 3.1.2-LTS.3支持的组件信息如下：
  - 分析集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez
  - 流式集群包含的组件有：Kafka,Flume,ZooKeeper,Ranger
  - 混合集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez,Kafka,Flume
  - 自定义集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Flume,Flink,Oozie,ZooKeeper,HetuEngine,Ranger,Tez,ClickHouse
- MRS 3.1.0支持的组件信息如下：
  - 分析集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,Kudu
  - 流式集群包含的组件有：Kafka,Flume,ZooKeeper,Ranger
  - 混合集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,Kudu,Kafka,Flume

  - 自定义集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Kafka,Flume,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,ClickHouse,Kudu
- MRS 3.0.5支持的组件信息如下：
  - 分析集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,Kudu,Alluxio
  - 流式集群包含的组件有：Kafka,Storm,Flume,ZooKeeper,Ranger
  - 混合集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,Kudu,Alluxio,Kafka,Storm,Flume
  - 自定义集群包含的组件有：Hadoop,Spark2x,HBase,Hive,Hue,Loader,Kafka,Storm,Flume,Flink,Oozie,ZooKeeper,Ranger,Tez,Impala,Presto,ClickHouse,Kudu,Alluxio
- MRS 2.1.0支持的组件信息如下：
  - 分析集群包含的组件有：Presto,Hadoop,Spark,HBase,Hive,Hue,Loader,Tez,Flink,Impala,Kudu
  - 流式集群包含的组件有：Kafka,Storm,Flume
- MRS 1.9.2支持的组件信息如下：
  - 分析集群包含的组件有：Presto,Hadoop,Spark,HBase,Opentsdb,Hive,Hue,Loader,Tez,Flink,Alluxio,Ranger
  - 流式集群包含的组件有：Kafka,KafkaManager,Storm,Flume

获取集群 ID

在调用作业相关接口的时候，部分URL中需要填入集群ID（cluster_id），所以需要先在管理控制台上获取到集群ID。集群ID获取步骤如下：

1. 登录MRS管理控制台。
2. 选择“现有集群”，单击待操作集群的集群名称，进入集群详情页面。
3. 选择“概览”页签，在基本信息区域获取“集群ID”。

图 10-3 集群 ID

获取作业 ID

在调用作业相关接口的时候，部分URL中需要填入作业ID（job_execution_id），所以需要先在管理控制台上获取到作业ID。作业ID获取步骤如下：

1. 登录MRS管理控制台。
2. 选择“现有集群”，单击待操作集群的集群名称，进入集群详情页面。
3. 选择“作业管理”页签，在作业列表中获取待操作作业对应的“ID”。

图 10-4 作业 ID

10.8 MRS 支持的角色与组件对应表

表 10-10 MRS 支持的角色与组件对应表

角色名	组件名
OMSServer	OMSServer
NameNode	HDFS
Zkfc	HDFS
JournalNode	HDFS
DataNode	HDFS
ResourceManager	Yarn
NodeManager	Yarn
JobHistoryServer	Mapreduce
quorumpeer	ZooKeeper
HMaster	HBase
ThriftServer	HBase
RegionServer	HBase
SlapdServer	LdapServer
KerberosServer	KrbServer

角色名	组件名
KerberosAdmin	KrbServer
Hue	Hue
LoaderServer	Loader
JDBCServer	Spark
JobHistory	Spark
SparkResource	Spark
JDBCServer2x	Spark2x
JobHistory2x	Spark2x
SparkResource2x	Spark2x
MetaStore	Hive
WebHCat	Hive
HiveServer	Hive
MonitorServer	Flume
Flume	Flume
oozie	Oozie
KerberosClient	KrbClient
SlapdClient	LdapClient
meta	meta
DBServer	DBService
Broker	Kafka
Supervisor	Storm
Logviewer	Storm
Nimbus	Storm
UI	Storm
FlinkResource	Flink
ClickHouseServer	ClickHouse
ClickHouseBalancer	ClickHouse
HSBroker	HetuEngine
HSConsole	HetuEngine
QAS	HetuEngine

角色名	组件名
CDLConnector	CDL
CDLService	CDL
IoTDBServer	IoTDB
ConfigNode	IoTDB
FE	Doris
BE	Doris
DBroker	Doris
TokenServer	Guardian
JobServer	JobGateway
JobBalancer	JobGateway