

云备份

API 参考

文档版本

03

发布日期

2023-06-29



版权所有 © 华为技术有限公司 2023。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目 录

1 使用前必读.....	1
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束限制.....	1
1.5 基本概念.....	2
2 API 概览.....	4
3 如何调用 API.....	5
3.1 构造请求.....	5
3.2 认证鉴权.....	9
3.3 返回结果.....	10
4 API.....	12
4.1 任务.....	12
4.1.1 查询单个任务.....	12
4.1.2 查询任务列表.....	22
4.2 可保护性.....	39
4.2.1 查询可保护资源.....	39
4.2.2 查询 agent 状态.....	52
4.2.3 查询指定可保护资源.....	57
4.2.4 查询复制能力.....	69
4.3 备份.....	72
4.3.1 同步备份.....	72
4.3.2 查询指定备份.....	78
4.3.3 查询所有备份.....	86
4.3.4 删除备份.....	98
4.3.5 复制备份.....	101
4.3.6 备份恢复.....	107
4.3.7 查询备份元数据.....	112
4.3.8 更新备份.....	117
4.4 备份共享.....	125
4.4.1 添加备份成员.....	126
4.4.2 更新备份成员状态.....	130

4.4.3 获取备份成员详情.....	135
4.4.4 获取备份成员列表.....	139
4.4.5 删除指定备份成员.....	144
4.5 存储库.....	147
4.5.1 创建存储库.....	148
4.5.2 查询指定存储库.....	167
4.5.3 查询存储库列表.....	179
4.5.4 修改存储库.....	189
4.5.5 删除存储库.....	200
4.5.6 移除资源.....	203
4.5.7 添加资源.....	207
4.5.8 设置存储库策略.....	212
4.5.9 解除存储库策略.....	217
4.5.10 查询其他区域存储库列表.....	221
4.5.11 迁移资源.....	230
4.5.12 批量修改存储库.....	234
4.5.13 设置存储库资源.....	239
4.5.14 创建包周期存储库.....	243
4.5.15 存储库容量总览.....	259
4.6 文件应用备份.....	263
4.6.1 新增备份路径.....	263
4.6.2 查询指定客户端.....	267
4.6.3 修改客户端.....	271
4.6.4 注册客户端.....	277
4.6.5 移除客户端.....	282
4.6.6 移除备份路径.....	285
4.6.7 查询客户端列表.....	290
4.7 标签.....	295
4.7.1 查询存储库资源实例.....	295
4.7.2 批量添加删除存储库资源标签.....	308
4.7.3 添加存储库资源标签.....	314
4.7.4 删除存储库资源标签.....	319
4.7.5 查询存储库资源标签.....	322
4.7.6 查询存储库项目标签.....	326
4.8 策略.....	330
4.8.1 创建策略.....	330
4.8.2 查询单个策略.....	342
4.8.3 查询策略列表.....	348
4.8.4 修改策略.....	355
4.8.5 删除策略.....	367
4.9 计量.....	370
4.9.1 查询容量统计.....	370

4.10 运营.....	374
4.10.1 变更.....	375
4.11 还原点.....	380
4.11.1 同步备份还原点.....	380
4.11.2 查询备份还原点.....	384
4.11.3 复制备份还原点.....	388
4.11.4 创建备份还原点.....	394
4.12 项目.....	403
4.12.1 查询租户的项目信息.....	403
4.12.2 查询租户项目列表.....	407
4.12.3 查询租户信息.....	411
4.12.4 租户迁移.....	414
4.12.5 查询迁移.....	418
5 应用示例.....	423
5.1 示例 1：创建弹性云服务器备份.....	423
5.2 示例 2：创建存储库自动备份.....	427
5.3 示例 3：查询备份信息.....	428
6 权限和授权项.....	431
6.1 权限及授权项说明.....	431
6.2 CBR 授权项分类.....	432
7 附录.....	439
7.1 状态码.....	439
7.2 错误码.....	440
7.3 获取项目 ID.....	453
A 修订记录.....	455

1 使用前必读

1.1 概述

欢迎使用云备份（Cloud Backup and Recovery，CBR）。云备份（Cloud Backup and Recovery）为云内的弹性云服务器、裸金属服务器、云硬盘，云下VMware虚拟化环境，提供简单易用的备份服务，针对病毒入侵、人为误删除、软硬件故障等场景，可将数据恢复到任意备份点。云备份可以为云内资源做云内备份/云内恢复；也可以将云下资源备份上云，云下/云内恢复。

您可以使用本文档提供的API对云备份进行相关操作，如创建存储库、删除存储库、创建策略、复制备份等。支持的全部操作请参见[API概览](#)。

在调用云备份API之前，请确保已经充分了解云备份相关概念，详细信息请参见[产品介绍](#)。

1.2 调用说明

云备份提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

1.3 终端节点

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

1.4 约束限制

您能创建的云备份资源的数量与配额有关系，如果您想查看服务配额、扩大配额，具体请参见“[配额](#)”。

更详细的限制请参见具体API的说明。

1.5 基本概念

- 账号

用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用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，您可以查看账号ID和IAM用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

从地理位置和网络时延维度划分，同一个Region内共享弹性计算、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](#)。

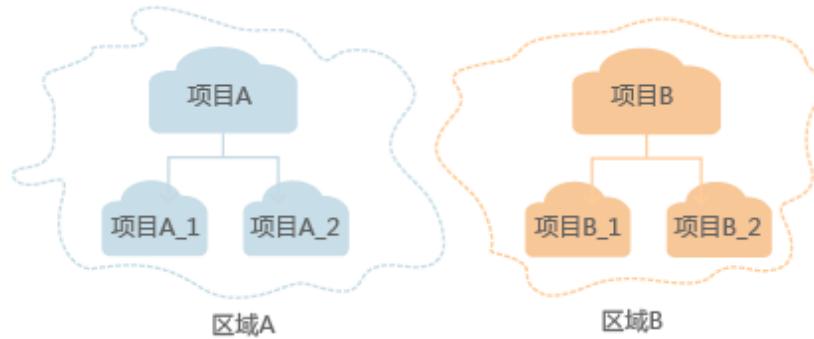
- 可用区（AZ, Availability Zone）

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

- 项目

区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，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的资源，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的资源进行分组和管理，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《[企业管理用户指南](#)》。

2 API 概览

云备份所提供的接口均为CBR接口，您可以通过这些接口完整的使用云备份的所有功能。

表 2-1 接口说明

类型	说明
任务	可以查询任务列表和单个任务的情况。
可保护性	可以查询所在区域的复制能力，本区域是否支持复制备份和存储库。
存储库	可以实现创建存储库、存储库查询、为存储库绑定策略等操作。
备份共享	用户可以将备份共享给其他用户使用。通过备份共享的接口完成备份共享的相关操作。
还原点	可以实现对存储库执行备份、执行复制的操作，以及查询备份和复制创建的时间。
备份	可以实现查询备份、同步备份副本、使用备份恢复数据等操作。
策略	绑定策略的存储库可以定期执行备份。通过策略相关接口完成创建策略、修改策略、查询策略等操作。
标签	可对存储库进行标签添加、编辑或删除操作。此处的标签仅用于存储库的过滤和管理。

3 如何调用 API

3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的[获取用户Token](#)来说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

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

请求 URI

请求URI由如下部分组成：

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

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

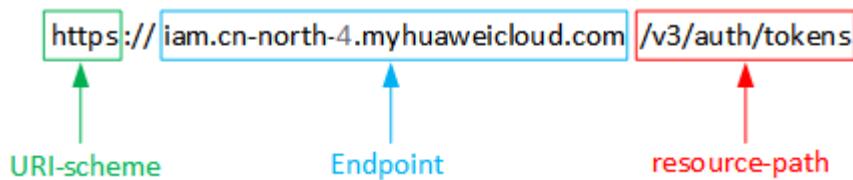
表 3-1 URI 中的参数说明

参数	描述
URI-scheme	表示用于传输请求的协议，当前所有API均采用 HTTPS 协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从 地区和终端节点 获取。 例如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，即API访问路径。从具体API的URI模块获取，例如“ 获取用户Token ”API的resource-path为“/v3/auth/tokens”。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“?”，形式为“参数名=参数取值”，例如“?limit=10”，表示查询不超过10条数据。

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

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

图 3-1 URI 示意图



说明

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

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务您正在请求什么类型的操作。

表 3-2 HTTP 方法

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

在[获取用户Token](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

`POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens`

请求消息头

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

详细的公共请求消息头字段请参见[表3-3](#)。

表 3-3 公共请求消息头

名称	描述	是否必选	示例
Host	请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。	否 使用AK/SK认证时该字段必选。	code.test.com or code.test.com:443
Content-Type	消息体的类型（格式）。推荐用户使用默认值application/json，有其他取值时会在具体接口中专门说明。	是	application/json
Content-Length	请求body长度，单位为Byte。	否	3495
X-Project-Id	project id，项目编号。请参考 获取项目ID 章节获取项目编号。	否 如果是专属云场景采用AK/SK认证方式的接口请求，或者多project场景采用AK/SK认证的接口请求，则该字段必选。	e9993fc787d94b6c886cb aa340f9c0f4
X-Auth-Token	用户Token。 用户Token也就是调用 获取用户Token 接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头(Headers)中包含的“X-Subject-Token”的值即为Token值。	否 使用Token认证时该字段必选。	注：以下仅为Token示例片段。 MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

□ 说明

API同时支持使用AK/SK认证，AK/SK认证使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

AK/SK认证的详细说明请参见[认证鉴权](#)的“AK/SK认证”。

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

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

请求消息体（可选）

该部分可选。请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-Type对应，传递除请求消息头之外的内容。若请求消息体中的参数支持中文，则中文字符必须为UTF-8编码。

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

对于[获取用户Token](#)接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写，其中**username**为用户名，**domainname**为用户所属的账号名称，*****为用户登录密码，xxxxxxxxxxxxxxxxxxxx为project的名称，如“cn-north-1”，您可以从[地区和终端节点](#)获取。

□ 说明

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

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

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

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于[获取用户Token](#)接口，返回的响应消息头中的

“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

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

Token 认证

说明

Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

Token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加到请求消息头中，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需要项目级别的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", //IAM用户名  
                    "password": "*****", //IAM用户密码  
                    "domain": {  
                        "name": "domainname" //IAM用户所属帐号名  
                    }  
                }  
            }  
        },  
        "scope": {  
            "project": {  
                "name": "xxxxxxxx" //项目名称  
            }  
        }  
    }  
}
```

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-1.myhuaweicloud.com/v3/auth/projects  
Content-Type: application/json  
X-Auth-Token: ABCDEFJ....
```

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

AK/SK 认证

□ 说明

AK/SK签名认证方式仅支持消息体大小在12MB以内，12MB以上的请求请使用Token认证。

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)：私有访问密钥。与访问密钥ID结合使用，对请求进行加密签名，可标识发送方，并防止请求被修改。

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

□ 说明

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。

3.3 返回结果

状态码

请求发送以后，您会收到响应，其中包含状态码、响应消息头和消息体。

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

对于[获取用户Token](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

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

图 3-2 获取用户 Token 响应消息头

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopener
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token
→ MIIXQVJKoZhvNAQcC0IYTjCCGEoCAQEExDTALBglghkgBZQMEAqEwgharBgkqhkiG9w0BBwGgg hacBIIWmHsidG9rZW4iOnsiZXhwaxJlc19hdCI6ljlwMTktMDItMTNUMDfj3KUs6YgKnpVNRbW2eZ5eb78SZOkqjACgkIqO1wi4JlGzrp d18LGXK5bxldfq4lqHCYb8P4NaY0NYejcAgzJveFIYtLWT1GSO0zxKZmlQHQj82H8qHdg lZO9fuEbL5dMhdavj+33wElxHRC9187o+k9-j+CMZSEB7bUGd5Uj6eRASX1jipPEGA270g1Fr uoL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUvHvpxk8pxiX1wTEboX-RzT6MUbpvGw-oPNFYxJECKnH3HRozv0vN--n5d6Nb xg==

x-xss-protection → 1; mode=block;
```

响应消息体（可选）

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[获取用户Token](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "az-01",
            ....

```

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

```
{
  "error_msg": "The format of message is error",
  "error_code": "AS.0001"
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 任务

4.1.1 查询单个任务

功能介绍

根据指定任务ID查询任务

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/operation-logs/{operation_log_id}

表 4-1 路径参数

参数	是否必选	参数类型	描述
operation_log_id	是	String	任务ID
project_id	是	String	项目ID

请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-3 响应 Body 参数

参数	参数类型	描述
operation_log	OperationLog object	任务信息

表 4-4 OperationLog

参数	参数类型	描述
checkpoint_id	String	备份记录id
created_at	String	创建时间,例如: "2020-02-23T01:00:32Z"
ended_at	String	任务结束时间,例如: "2020-02-23T01:00:32Z"
error_info	OpErrorInfo object	任务错误信息
extra_info	OpExtraInfo object	任务扩展信息
id	String	任务id

参数	参数类型	描述
operation_type	String	任务类型 枚举值： <ul style="list-style-type: none">• backup• copy• replication• restore• delete• sync• vault_delete• remove_resource
policy_id	String	策略ID
project_id	String	项目ID
provider_id	String	备份提供商ID。用于区分备份对象。
started_at	String	任务开始时间,例如: "2020-02-23T01:00:32Z"
status	String	任务状态 枚举值： <ul style="list-style-type: none">• success• skipped• failed• running• timeout• waiting
updated_at	String	修改时间,例如: "2020-02-23T01:00:32Z"
vault_id	String	任务操作资源所属存储库ID
vault_name	String	任务操作资源所属存储库名称

表 4-5 OpErrorInfo

参数	参数类型	描述
code	String	请参见 错误码 。
message	String	错误信息

表 4-6 OpExtraInfo

参数	参数类型	描述
backup	OpExtendInfoBackup object	备份扩展参数
common	OpExtendInfoCommon object	公共参数
delete	OpExtendInfoDelete object	删除扩展参数
sync	OpExtendInfoSync object	扩展同步信息
remove_resources	OpExtendInfoRemoveResources object	移除Vault的资源
replication	OpExtendInfoReplication object	扩展复制信息
resource	Resource object	资源信息
restore	OpExtendInfoRestore object	扩展恢复信息
vault_delete	OpExtendInfoVaultDelete object	删除vault

表 4-7 OpExtendInfoBckup

参数	参数类型	描述
app_consistency_error_code	String	应用一致性备份失败错误码。请参见 错误码 。
app_consistency_error_message	String	应用一致性备份错误信息

参数	参数类型	描述
app_consistency_status	String	应用一致性备份状态；0:非应用一致性，1：应用一致性备份 枚举值： <ul style="list-style-type: none">• 0• 1
backup_id	String	备份副本ID
backup_name	String	备份名称
incremental	String	是否增备 枚举值： <ul style="list-style-type: none">• "true"• "false"

表 4-8 OpExtendInfoCommon

参数	参数类型	描述
progress	Integer	进度，取值为0-100 最小值：0 最大值：100
request_id	String	请求id
task_id	String	备份任务id

表 4-9 OpExtendInfoDelete

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称

表 4-10 OpExtendInfoSync

参数	参数类型	描述
sync_backup_num	Integer	同步备份副本数
delete_backup_num	Integer	删除的备份副本数

参数	参数类型	描述
err_sync_backup_num	Integer	同步失败备份副本数

表 4-11 OpExtendInfoRemoveResources

参数	参数类型	描述
fail_count	Integer	删除失败的资源数量
total_count	Integer	删除的备份数量
resources	Array of Resource objects	资源信息

表 4-12 Resource

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称，长度限制：0-255 最小长度： 0 最大长度： 255
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-13 OpExtendInfoReplication

参数	参数类型	描述
destination_backup_id	String	目标副本ID
destination_checkpoint_id	String	目标还原点ID
destination_project_id	String	目标project_id

参数	参数类型	描述
destination_region	String	目标区域
source_backup_id	String	源副本ID
source_checkpoint_id	String	源还原点ID
source_project_id	String	源project_id
source_region	String	源区域
source_backup_name	String	源备份名称
destination_backup_name	String	目标备份名称

表 4-14 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-15 OpExtendInfoRestore

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称
target_resource_id	String	恢复目标资源ID
target_resource_name	String	恢复目标资源名称

表 4-16 OpExtendInfoVaultDelete

参数	参数类型	描述
fail_count	Integer	本次任务删除失败的资源数量
total_count	Integer	本次任务删除的备份总数

状态码： 404

表 4-17 感应 Body 参数

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

请求示例

GET https://{endpoint}/v3/{project_id}/operation-logs/{operation_log_id}

响应示例

状态码： 200

OK

```
{  
    "operation_log": {  
        "status": "success",  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "checkpoint_id": "b432511f-d889-428f-8b0e-5f47c524c6b6",  
        "updated_at": "2019-05-23T14:35:23.584+00:00",  
        "error_info": {  
            "message": "",  
            "code": ""  
        },  
        "started_at": "2019-05-23T14:31:36.007+00:00",  
        "id": "4827f2da-b008-4507-ab7d-42d0df5ed912",  
        "extra_info": {  
            "resource": {  
                "type": "OS::Nova::Server",  
                "id": "1dab32fa-ebf2-415a-ab0b-eabe6353bc86",  
                "name": "ECS-0001"  
            },  
            "backup": {  
                "backup_name": "manualbk_1234",  
                "backup_id": "0e5d0ef6-7f0a-4890-b98c-cb12490e31c1"  
            },  
            "common": {  
                "progress": 100,  
                "request_id": "req-cdb98cc4-e87b-4f40-9b4a-57ec036620bc"  
            }  
        },  
        "ended_at": "2019-05-23T14:35:23.511+00:00",  
        "created_at": "2019-05-23T14:31:36.039+00:00",  
        "operation_type": "backup",  
        "project_id": "04f1829c788037ac2fb8c01eb2b04b95"  
    }  
}
```

状态码： 404

任务ID不存在

```
{  
    "error_code": "BackupService.6500",  
}
```

```
        "error_msg" : "Operation log does not exist."  
    }
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowOpLogSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowOpLogRequest request = new ShowOpLogRequest();  
        try {  
            ShowOpLogResponse response = client.showOpLog(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	OK
404	任务ID不存在

错误码

请参见[错误码](#)。

4.1.2 查询任务列表

功能介绍

查询任务列表

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/operation-logs

表 4-18 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-19 Query 参数

参数	是否必选	参数类型	描述
end_time	否	String	任务结束时间，格式为%YYYY-%mm-%ddT%HH:%MM:%SSZ，例如2018-02-01T12:00:00Z
enterprise_project_id	否	String	企业项目id或all_granted_eps，all_granted_eps表示查询用户有权限的所有企业项目id

参数	是否必选	参数类型	描述
limit	否	Integer	每页显示的条目数量，正整数 最小值：1
offset	否	Integer	偏移值，正整数 最小值：1
operation_type	否	String	任务类型 枚举值： <ul style="list-style-type: none">• backup• copy• replication• delete• restore• vault_delete• remove_resource• sync
provider_id	否	String	备份提供商ID
resource_id	否	String	该任务操作的资源ID
resource_name	否	String	该任务操作的资源名称
start_time	否	String	任务开始时间，格式为%YYYY-%mm-%ddT%HH:%MM:%SSZ，例如2018-01-01T12:00:00Z
status	否	String	任务状态 枚举值： <ul style="list-style-type: none">• success• skipped• failed• running• timeout• waiting
vault_id	否	String	存储库ID,该任务操作的资源所属绑定的存储库。
vault_name	否	String	存储库名称，该任务操作资源所绑定的存储库名称。

请求参数

表 4-20 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-21 响应 Body 参数

参数	参数类型	描述
operation_logs	Array of OperationLog objects	任务列表
count	Integer	任务个数
limit	Integer	每页显示的条目数量 最小值： 1 最大值： 1000 缺省值： 1000
offset	Integer	偏移量，表示从此偏移量开始查询 最小值： 0 缺省值： 0

表 4-22 OperationLog

参数	参数类型	描述
checkpoint_id	String	备份记录id
created_at	String	创建时间,例如: "2020-02-23T01:00:32Z"
ended_at	String	任务结束时间,例如: "2020-02-23T01:00:32Z"
error_info	OpErrorInfo object	任务错误信息
extra_info	OpExtraInfo object	任务扩展信息

参数	参数类型	描述
id	String	任务id
operation_type	String	任务类型 枚举值： <ul style="list-style-type: none">• backup• copy• replication• restore• delete• sync• vault_delete• remove_resource
policy_id	String	策略ID
project_id	String	项目ID
provider_id	String	备份提供商ID。用于区分备份对象。
started_at	String	任务开始时间,例如: "2020-02-23T01:00:32Z"
status	String	任务状态 枚举值： <ul style="list-style-type: none">• success• skipped• failed• running• timeout• waiting
updated_at	String	修改时间,例如: "2020-02-23T01:00:32Z"
vault_id	String	任务操作资源所属存储库ID
vault_name	String	任务操作资源所属存储库名称

表 4-23 OpErrorInfo

参数	参数类型	描述
code	String	请参见 错误码 。
message	String	错误信息

表 4-24 OpExtraInfo

参数	参数类型	描述
backup	OpExtendInfoBackup object	备份扩展参数
common	OpExtendInfoCommon object	公共参数
delete	OpExtendInfoDelete object	删除扩展参数
sync	OpExtendInfoSync object	扩展同步信息
remove_resources	OpExtendInfoRemoveResources object	移除Vault的资源
replication	OpExtendInfoReplication object	扩展复制信息
resource	Resource object	资源信息
restore	OpExtendInfoRestore object	扩展恢复信息
vault_delete	OpExtendInfoVaultDelete object	删除vault

表 4-25 OpExtendInfoBackup

参数	参数类型	描述
app_consistency_error_code	String	应用一致性备份失败错误码。请参见 错误码 。
app_consistency_error_message	String	应用一致性备份错误信息

参数	参数类型	描述
app_consistency_status	String	应用一致性备份状态；0:非应用一致性，1：应用一致性备份 枚举值： <ul style="list-style-type: none">• 0• 1
backup_id	String	备份副本ID
backup_name	String	备份名称
incremental	String	是否增备 枚举值： <ul style="list-style-type: none">• "true"• "false"

表 4-26 OpExtendInfoCommon

参数	参数类型	描述
progress	Integer	进度，取值为0-100 最小值：0 最大值：100
request_id	String	请求id
task_id	String	备份任务id

表 4-27 OpExtendInfoDelete

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称

表 4-28 OpExtendInfoSync

参数	参数类型	描述
sync_backup_num	Integer	同步备份副本数
delete_backup_num	Integer	删除的备份副本数

参数	参数类型	描述
err_sync_backup_num	Integer	同步失败备份副本数

表 4-29 OpExtendInfoRemoveResources

参数	参数类型	描述
fail_count	Integer	删除失败的资源数量
total_count	Integer	删除的备份数量
resources	Array of Resource objects	资源信息

表 4-30 Resource

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称，长度限制：0-255 最小长度： 0 最大长度： 255
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-31 OpExtendInfoReplication

参数	参数类型	描述
destination_backup_id	String	目标副本ID
destination_checkpoint_id	String	目标还原点ID
destination_project_id	String	目标project_id

参数	参数类型	描述
destination_region	String	目标区域
source_backup_id	String	源副本ID
source_checkpoint_id	String	源还原点ID
source_project_id	String	源project_id
source_region	String	源区域
source_backup_name	String	源备份名称
destination_backup_name	String	目标备份名称

表 4-32 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-33 OpExtendInfoRestore

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称
target_resource_id	String	恢复目标资源ID
target_resource_name	String	恢复目标资源名称

表 4-34 OpExtendInfoVaultDelete

参数	参数类型	描述
fail_count	Integer	本次任务删除失败的资源数量
total_count	Integer	本次任务删除的备份总数

状态码： 404

表 4-35 响应 Body 参数

参数	参数类型	描述
operation_logs	Array of OperationLog objects	任务列表
count	Integer	任务个数
limit	Integer	每页显示的条目数量 最小值: 1 最大值: 1000 缺省值: 1000
offset	Integer	偏移量, 表示从此偏移量开始查询 最小值: 0 缺省值: 0

表 4-36 OperationLog

参数	参数类型	描述
checkpoint_id	String	备份记录id
created_at	String	创建时间,例如: "2020-02-23T01:00:32Z"
ended_at	String	任务结束时间,例如: "2020-02-23T01:00:32Z"
error_info	OpErrorInfo object	任务错误信息
extra_info	OpExtraInfo object	任务扩展信息
id	String	任务id

参数	参数类型	描述
operation_type	String	任务类型 枚举值： <ul style="list-style-type: none">• backup• copy• replication• restore• delete• sync• vault_delete• remove_resource
policy_id	String	策略ID
project_id	String	项目ID
provider_id	String	备份提供商ID。用于区分备份对象。
started_at	String	任务开始时间,例如: "2020-02-23T01:00:32Z"
status	String	任务状态 枚举值： <ul style="list-style-type: none">• success• skipped• failed• running• timeout• waiting
updated_at	String	修改时间,例如: "2020-02-23T01:00:32Z"
vault_id	String	任务操作资源所属存储库ID
vault_name	String	任务操作资源所属存储库名称

表 4-37 OpErrorInfo

参数	参数类型	描述
code	String	请参见 错误码 。
message	String	错误信息

表 4-38 OpExtraInfo

参数	参数类型	描述
backup	OpExtendInfoBackup object	备份扩展参数
common	OpExtendInfoCommon object	公共参数
delete	OpExtendInfoDelete object	删除扩展参数
sync	OpExtendInfoSync object	扩展同步信息
remove_resources	OpExtendInfoRemoveResources object	移除Vault的资源
replication	OpExtendInfoReplication object	扩展复制信息
resource	Resource object	资源信息
restore	OpExtendInfoRestore object	扩展恢复信息
vault_delete	OpExtendInfoVaultDelete object	删除vault

表 4-39 OpExtendInfoBackup

参数	参数类型	描述
app_consistency_error_code	String	应用一致性备份失败错误码。请参见 错误码 。
app_consistency_error_message	String	应用一致性备份错误信息

参数	参数类型	描述
app_consistency_status	String	应用一致性备份状态；0:非应用一致性，1：应用一致性备份 枚举值： <ul style="list-style-type: none">• 0• 1
backup_id	String	备份副本ID
backup_name	String	备份名称
incremental	String	是否增备 枚举值： <ul style="list-style-type: none">• "true"• "false"

表 4-40 OpExtendInfoCommon

参数	参数类型	描述
progress	Integer	进度，取值为0-100 最小值：0 最大值：100
request_id	String	请求id
task_id	String	备份任务id

表 4-41 OpExtendInfoDelete

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称

表 4-42 OpExtendInfoSync

参数	参数类型	描述
sync_backup_num	Integer	同步备份副本数
delete_backup_num	Integer	删除的备份副本数

参数	参数类型	描述
err_sync_backup_num	Integer	同步失败备份副本数

表 4-43 OpExtendInfoRemoveResources

参数	参数类型	描述
fail_count	Integer	删除失败的资源数量
total_count	Integer	删除的备份数量
resources	Array of Resource objects	资源信息

表 4-44 Resource

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称，长度限制：0-255 最小长度： 0 最大长度： 255
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-45 OpExtendInfoReplication

参数	参数类型	描述
destination_backup_id	String	目标副本ID
destination_checkpoint_id	String	目标还原点ID
destination_project_id	String	目标project_id

参数	参数类型	描述
destination_region	String	目标区域
source_backup_id	String	源副本ID
source_checkpoint_id	String	源还原点ID
source_project_id	String	源project_id
source_region	String	源区域
source_backup_name	String	源备份名称
destination_backup_name	String	目标备份名称

表 4-46 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-47 OpExtendInfoRestore

参数	参数类型	描述
backup_id	String	备份副本ID
backup_name	String	备份名称
target_resource_id	String	恢复目标资源ID
target_resource_name	String	恢复目标资源名称

表 4-48 OpExtendInfoVaultDelete

参数	参数类型	描述
fail_count	Integer	本次任务删除失败的资源数量
total_count	Integer	本次任务删除的备份总数

请求示例

GET https://[endpoint]/v3/{project_id}/operation-logs

响应示例

状态码： 200

OK

```
{  
    "count" : 1,  
    "operation_logs" : [ {  
        "status" : "success",  
        "provider_id" : "0daac4c5-6707-4851-97ba-169e36266b66",  
        "checkpoint_id" : "b432511f-d889-428f-8b0e-5f47c524c6b6",  
        "updated_at" : "2019-05-23T14:35:23.584+00:00",  
        "error_info" : {  
            "message" : "",  
            "code" : ""  
        },  
        "started_at" : "2019-05-23T14:31:36.007+00:00",  
        "id" : "4827f2da-b008-4507-ab7d-42d0df5ed912",  
        "extra_info" : {  
            "resource" : {  
                "type" : "OS::Nova::Server",  
                "id" : "1dab32fa-ebf2-415a-ab0b-eabe6353bc86",  
                "name" : "ECS-0001"  
            },  
            "backup" : {  
                "backup_name" : "manualbk_backup",  
                "backup_id" : "0e5d0ef6-7f0a-4890-b98c-cb12490e31c1"  
            },  
            "common" : {  
                "progress" : 100,  
                "request_id" : "req-cdb98cc4-e87b-4f40-9b4a-57ec036620bc"  
            }  
        },  
        "ended_at" : "2019-05-23T14:35:23.511+00:00",  
        "created_at" : "2019-05-23T14:31:36.039+00:00",  
        "operation_type" : "backup",  
        "project_id" : "04f1829c788037ac2fb8c01eb2b04b95"  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;
```

```
public class ListOpLogsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListOpLogsRequest request = new ListOpLogsRequest();  
        request.withEndTime("<end_time>");  
        request.withLimit(<limit>);  
        request.withOffset(<offset>);  
        request.withOperationType(ListOpLogsRequest.OperationTypeEnum.fromValue("<operation_type>"));  
        request.withProviderId("<provider_id>");  
        request.withResourceId("<resource_id>");  
        request.withResourceName("<resource_name>");  
        request.withStartTime("<start_time>");  
        request.withStatus(ListOpLogsRequest.StatusEnum.fromValue("<status>"));  
        request.withVaultId("<vault_id>");  
        request.withVaultName("<vault_name>");  
        request.withEnterpriseProjectId("<enterprise_project_id>");  
        try {  
            ListOpLogsResponse response = client.listOpLogs(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatus());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \\\
```

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

try:
    request = ListOpLogsRequest()
    request.end_time = "<end_time>"
    request.limit = <limit>
    request.offset = <offset>
    request.operation_type = "<operation_type>"
    request.provider_id = "<provider_id>"
    request.resource_id = "<resource_id>"
    request.resource_name = "<resource_name>"
    request.start_time = "<start_time>"
    request.status = "<status>"
    request.vault_id = "<vault_id>"
    request.vault_name = "<vault_name>"
    request.enterprise_project_id = "<enterprise_project_id>"
    response = client.list_op_logs(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListOpLogsRequest{}
    endTimeRequest:= "<end_time>"
    request.EndTime = &endTimeRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    operationTypeRequest:= model.GetListOpLogsRequestOperationTypeEnum().<OPERATION_TYPE>
    request.OperationType = &operationTypeRequest
    providerIdRequest:= "<provider_id>"
```

```
request.ProviderId = &providerIdRequest
resourceIdRequest:= "<resource_id>"
request.ResourceId = &resourceIdRequest
resourceNameRequest:= "<resource_name>"
request.ResourceName = &resourceNameRequest
startTimeRequest:= "<start_time>"
request.StartTime = &startTimeRequest
statusRequest:= model.GetListOpLogsRequestStatusEnum().<STATUS>
request.Status = &statusRequest
vaultIdRequest:= "<vault_id>"
request.VaultId = &vaultIdRequest
vaultNameRequest:= "<vault_name>"
request.VaultName = &vaultNameRequest
enterpriseProjectIdRequest:= "<enterprise_project_id>"
request.EnterpriseProjectId = &enterpriseProjectIdRequest
response, err := client.ListOpLogs(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK
404	Not Found

错误码

请参见[错误码](#)。

4.2 可保护性

4.2.1 查询可保护资源

功能介绍

查询可保护性资源列表

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/protectables/{protectable_type}/instances

表 4-49 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
protectable_type	是	String	对象类型 枚举值： <ul style="list-style-type: none">• server• disk

表 4-50 Query 参数

参数	是否必选	参数类型	描述
id	否	String	根据资源id过滤
limit	否	Integer	每页显示的条目数量，每页最多支持50条
marker	否	String	上一次查询最后一条的ID
name	否	String	按名称过滤
offset	否	Integer	偏移值
server_id	否	String	根据该id过滤属于该服务器的所有磁盘，支持企业多项目的用户才能传入此参数
status	否	String	资源的状态，如available, error 等

请求参数

无

响应参数

状态码： 200

表 4-51 响应 Body 参数

参数	参数类型	描述
instances	Array of Protectables Resp objects	可保护性查询实例

表 4-52 ProtectablesResp

参数	参数类型	描述
children	Array of objects	子资源
detail	Object	资源详情
id	String	id
name	String	名称
protectable	ProtectableResult object	是否可备份
size	Integer	大小，单位GB
status	String	资源状态 枚举值： <ul style="list-style-type: none">• active• deleted• error
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-53 ProtectableResult

参数	参数类型	描述
code	String	不支持备份的错误码
reason	String	不支持备份的原因
result	Boolean	是否可备份
vault	VaultGet object	绑定的存储库
message	String	资源不可备份的原因信息，当资源可保护性检验失败时才有该字段。

表 4-54 VaultGet

参数	参数类型	描述
billing	Billing object	运营信息

参数	参数类型	描述
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	资源
tags	Array of Tag objects	存储库标签
enterprise_pro ject_id	String	企业项目id， 默认为 ‘0’ 。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRul es object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储 库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分 比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80
sys_lock_sourc e_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值： <ul style="list-style-type: none">• SMB• "

参数	参数类型	描述
updated_at	String	更新时间,例如:"2020-02-05T10:38:34.209782"
version	String	版本

表 4-55 Billing

参数	参数类型	描述
allocated	Integer	已分配容量, 单位GB
charging_mod e	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal
status	String	存储库状态 枚举值: <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名

参数	参数类型	描述
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-56 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值: <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小
backup_count	Integer	副本数量

表 4-57 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效, 排除不需要备份的磁盘。当虚拟机新绑定磁 盘时, 也能继续排除之前设置不用备份的卷。

表 4-58 Tag

参数	参数类型	描述
key	String	<p>键。</p> <p>key最大长度为36个字符。</p> <p>key不能为空字符串。</p> <p>key前后空格会被丢弃。</p> <p>key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。</p> <p>key只能由中文, 字母, 数字, “-” , “_” 组成。</p>
value	String	<p>值。</p> <p>添加标签时value值必选, 删除标签时value值可选。</p> <p>value最大长度为43个字符。</p> <p>value可以为空字符串。</p> <p>value前后的空格会被丢弃。</p> <p>value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。</p> <p>value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。</p>

表 4-59 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	<p>按tags过滤自动绑定的资源</p> <p>最小长度: 0</p> <p>最大长度: 5</p> <p>数组长度: 0 - 5</p>

表 4-60 BindRulesTags

参数	参数类型	描述
key	String	<p>key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。</p> <p>key只能由中文, 字母, 数字, “-” , “_” 组成。</p>

参数	参数类型	描述
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

```
GET https://{endpoint}/v3/{project_id}/protectables/{protectable_type}/instances
```

响应示例

状态码: 200

OK

```
{  
    "instances": [ {  
        "status": "ACTIVE",  
        "children": [ {  
            "status": "in-use",  
            "name": "no_delete_for_cbr_interface-volume-0000",  
            "detail": {  
                "attachments": [ {  
                    "server_id": "55ecd8b8-1457-4a2a-a9df-53756a690082",  
                    "attachment_id": "e16631fa-0522-4ac8-8f78-19db435caf68",  
                    "volume_id": "13eb535f-8635-4faf-9f08-584d21e78f0a",  
                    "attached_at": "2019-05-08T06:55:26.564+00:00",  
                    "device": "/dev/vda",  
                    "id": "13eb535f-8635-4faf-9f08-584d21e78f0a"  
                } ],  
                "links": [ {  
                    "href": "https://volume.region1.com/v2/3aec9116a6a5487ca14987121d2818fe/volumes/  
13eb535f-8635-4faf-9f08-584d21e78f0a",  
                    "rel": "self"  
                }, {  
                    "href": "https://volume.region1.com/3aec9116a6a5487ca14987121d2818fe/volumes/  
13eb535f-8635-4faf-9f08-584d21e78f0a",  
                    "rel": "bookmark"  
                } ],  
                "availability_zone": "AZ1",  
                "os-vol-host-attr:host": "AZ1.xxxxxx#kvm_fusionstorage",  
                "encrypted": false,  
                "updated_at": "2019-05-08T08:33:07.855+00:00",  
                "replication_status": "disabled",  
                "id": "13eb535f-8635-4faf-9f08-584d21e78f0a",  
                "size": 40,  
                "user_id": "3204777a731c4ebea330350c04f41b92",  
                "os-vol-tenant-attr:tenant_id": "3aec9116a6a5487ca14987121d2818fe",  
                "metadata": {  
                    "policy": "2ecbcffd-08ba-43f7-bf8d-95daba753464",  
                    "readonly": "False",  
                    "attached_mode": "rw"  
                },  
                "status": "in-use",  
                "volume_image_metadata": {  
                    "container_format": "bare",  
                    "min_ram": 0,  
                    "__account_code": "",  
                    "__os_bit": 64,  
                    "size": 2,  
                    "__os_version": "CentOS 7.2 64bit",  
                }  
            } ]  
        } ]  
    } ]  
}
```

```
"__backup_id" : "",  
"__description" : "",  
"__support_diskintensive" : "true",  
"disk_format" : "vhd",  
"__isregistered" : "true",  
"__image_size" : 3295645696,  
"__platform" : "CentOS",  
"__root_origin" : "file",  
"__originalimagename" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",  
"__quick_start" : "true",  
"image_name" : "CentOS7.2-cloudinit",  
"image_id" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",  
"__os_type" : "Linux",  
"__image_location" : "172.100.45.130:443:pcsimfs8bd00bf6c304d2e8cf4e2beb0aed702:703eb90a-a65f-4c72-b4c1-f070a55d068b",  
"min_disk" : 40,  
"__data_origin" : "file,jingxiang:CentOS-7.2-64bit.zvhdc2",  
"__support_kvm" : "true",  
"virtual_env_type" : "FusionCompute",  
"__image_source_type" : "uds",  
"__support_xen" : "true",  
"checksum" : "99914b932bd37a50b983c5e7c90ae93b",  
"__imagetype" : "gold",  
"__productcode" : "",  
},  
"multiattach" : false,  
"name" : "no_delete_for_cbr_interface-volume-0000",  
"bootable" : "true",  
"created_at" : "2019-05-08T06:54:58.478+00:00",  
"volume_type" : "SATA",  
"shareable" : false  
},  
"children" : [ ],  
"type" : "OS::Cinder::Volume",  
"id" : "13eb535f-8635-4faf-9f08-584d21e78f0a",  
"size" : 40  
}, {  
"status" : "in-use",  
"name" : "volume-f926-0001",  
"detail" : {  
"attachments" : [ {  
"server_id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",  
"attachment_id" : "3b32cb75-bd96-493b-8fb2-13f063c876d5",  
"volume_id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",  
"attached_at" : "2019-05-08T06:56:25.892+00:00",  
"device" : "/dev/vdb",  
"id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99"  
} ],  
"links" : [ {  
"href" : "https://volume.region1.com/v2/3aec9116a6a5487ca14987121d2818fe/volumes/8d3d519fd409-4be3-8a1f-9f7a5ecb7a99",  
"rel" : "self"  
}, {  
"href" : "https://volume.region1.com/3aec9116a6a5487ca14987121d2818fe/volumes/8d3d519fd409-4be3-8a1f-9f7a5ecb7a99",  
"rel" : "bookmark"  
},  
"availability_zone" : "AZ1",  
"os-vol-host-attr:host" : "AZ1.xxxxxx#FusionStorage",  
"encrypted" : false,  
"updated_at" : "2019-05-08T06:56:26.207+00:00",  
"replication_status" : "disabled",  
"id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",  
"size" : 10,  
"user_id" : "f3d66502790d4ddaaef5056059ed8f4",  
"os-vol-tenant-attr:tenant_id" : "3aec9116a6a5487ca14987121d2818fe",  
"metadata" : {  
"policy" : "cf3274fc-d23e-4c12-8be7-1e41b3c2faea",  
"readonly" : "False",
```

```
        "attached_mode" : "rw"
    },
    "status" : "in-use",
    "multiattach" : false,
    "name" : "volume-f926-0001",
    "bootable" : "false",
    "created_at" : "2019-03-26T09:30:46.784+00:00",
    "volume_type" : "SATA",
    "shareable" : false
},
"children" : [ ],
"type" : "OS::Cinder::Volume",
"id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",
"size" : 10
} ],
"protectable" : {
    "reason" : "",
    "code" : "",
    "result" : true
},
"name" : "no_delete_for_cbr_interface",
"detail" : {
    "addresses" : [
        "94b88790-c906-4450-b7c2-1c2b1ea639bf" : [ {
            "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:9b:9d:af",
            "version" : 4,
            "OS-EXT-IPS:port_id" : "f22f531a-d38a-480b-b59e-f7e5db0dcdcc",
            "addr" : "192.168.1.44",
            "OS-EXT-IPS:type" : "fixed"
        }, {
            "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:9b:9d:af",
            "version" : 4,
            "OS-EXT-IPS:port_id" : "f22f531a-d38a-480b-b59e-f7e5db0dcdcc",
            "addr" : "100.78.0.199",
            "OS-EXT-IPS:type" : "floating"
        } ]
    },
    "OS-EXT-SRV-ATTR:ramdisk_id" : "",
    "image" : "",
    "enterprise_project_id" : 0,
    "updated" : "2019-05-08T06:57:02.000+00:00",
    "OS-EXT-SRV-ATTR:user_data" :
    "lyEvYmluL2Jhc2gKZWNobyAncm9vdDokNiQ5c1FaeXkdnk5N1pGYXg5WUpRTUpPRElGdGdjYnovTUdEZ1dad
3E4VTM1dWE4MjQxTmtDTG1GRjJUUUm1BdnZSQnNCLy93VER0aFVRNUt0Z1p6dUlrenp2NmQwYTAnlHwgY2h
wYXNzd2QgLUU7",
    "OS-EXT-STS:vm_state" : "active",
    "OS-EXT-SRV-ATTR:instance_name" : "instance-0004456e",
    "OS-EXT-SRV-ATTR:root_device_name" : "/dev/vda",
    "OS-SRV-USG:launched_at" : "2019-05-08T06:55:38.000+00:00",
    "flavor" : {
        "vcpus" : 1,
        "disk" : 0,
        "ram" : 1024,
        "id" : "s3.small.1",
        "name" : "s3.small.1"
    },
    "id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",
    "security_groups" : [ {
        "name" : "default"
    } ],
    "os-extended-volumes:volumes_attached" : [ {
        "device" : "/dev/vda",
        "id" : "13eb535f-8635-4faf-9f08-584d21e78f0a",
        "delete_on_termination" : false,
        "bootIndex" : 0
    }, {
        "device" : "/dev/vdb",
        "id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",
        "delete_on_termination" : false
    }
]
```

```
        },
        "user_id" : "f3d66502790d4ddaaef5056059ed8f4",
        "OS-EXT-SRV-ATTR:hostname" : "no-delete-for-cbr-interface",
        "OS-DCF:diskConfig" : "MANUAL",
        "accessIPv4" : "",
        "accessIPv6" : "",
        "OS-EXT-SRV-ATTR:reservation_id" : "r-kpb5xckf",
        "progress" : 0,
        "OS-EXT-STS:power_state" : 1,
        "OS-EXT-AZ:availability_zone" : "AZ1",
        "metadata" : {
            "metering.image_id" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",
            "metering.imagetype" : "gold",
            "metering.resourcespeccode" : "s3.small.1.linux",
            "metering.cloudServiceType" : "sys.service.type.ec2",
            "virtual_env_type" : "FusionCompute",
            "image_name" : "CentOS7.2-cloudinit",
            "metering.resourctype" : 1,
            "vpc_id" : "94b88790-c906-4450-b7c2-1c2b1ea639bf",
            "os_bit" : 64,
            "cascaded.instance_extrainfo" : "",
            "os_type" : "Linux",
            "charging_mode" : 0
        },
        "status" : "ACTIVE",
        "sys_tags" : [ {
            "value" : 0,
            "key" : "_sys_enterprise_project_id"
        }],
        "os:scheduler_hints" : { },
        "hostId" : "179f22cafe6b951921bd6deb3b13a978df620427fb37bbb7c619de98",
        "OS-EXT-SRV-ATTR:host" : "AZ1.xxxxxx",
        "description" : "",
        "tags" : [ ],
        "OS-EXT-SRV-ATTR:kernel_id" : "",
        "locked" : false,
        "OS-EXT-SRV-ATTR:hypervisor_hostname" : "nova003@36",
        "name" : "no_delete_for_cbr_interface",
        "OS-EXT-SRV-ATTR:launch_index" : 0,
        "created" : "2019-05-08T06:55:22.000+00:00",
        "tenant_id" : "3aec9116a6a5487ca14987121d2818fe",
        "host_status" : "UP",
        "config_drive" : ""
    },
    "type" : "OS::Nova::Server",
    "id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",
    "size" : 50
}
]
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;
```

```
public class ListProtectableSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListProtectableRequest request = new ListProtectableRequest();  
        request.withLimit(<limit>);  
        request.withMarker("<marker>");  
        request.withName("<name>");  
        request.withOffset(<offset>);  
        request.withStatus("<status>");  
        request.withId("<id>");  
        request.withServerId("<server_id>");  
        try {  
            ListProtectableResponse response = client.listProtectable(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CbrClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CbrRegion.value_of("<YOUR REGION>")) \  
        .build()
```

```
try:  
    request = ListProtectableRequest()  
    request.limit = <limit>  
    request.marker = "<marker>"  
    request.name = "<name>"  
    request.offset = <offset>  
    request.status = "<status>"  
    request.id = "<id>"  
    request.server_id = "<server_id>"  
    response = client.list_protectable(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
    request := &model.ListProtectableRequest{}  
    limitRequest:= int32(<limit>)  
    request.Limit = &limitRequest  
    markerRequest:= "<marker>"  
    request.Marker = &markerRequest  
    nameRequest:= "<name>"  
    request.Name = &nameRequest  
    offsetRequest:= int32(<offset>)  
    request.Offset = &offsetRequest  
    statusRequest:= "<status>"  
    request.Status = &statusRequest  
    idRequest:= "<id>"  
    request.Id = &idRequest  
    serverIdRequest:= "<server_id>"  
    request.ServerId = &serverIdRequest  
    response, err := client.ListProtectable(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)
```

```
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.2.2 查询 agent 状态

功能介绍

检查应用一致性Agent状态

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/agent/check

表 4-61 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-62 请求 Body 参数

参数	是否必选	参数类型	描述
agent_status	是	Array of ProtectableAgentStatusResource objects	查询参数列表

表 4-63 ProtectableAgentStatusResource

参数	是否必选	参数类型	描述
resource_id	是	String	待检查资源ID
resource_name	否	String	待检查资源name
resource_type	是	String	待检查的资源类型。当前支持的取值包含两个： OS::Nova::Server，该值代表保护的资源为云服务器， OS::Ironic::BareMetalServer，该值代表保护的资源为裸金属服务器。

响应参数

状态码： 200

表 4-64 响应 Body 参数

参数	参数类型	描述
agent_status	Array of ProtectableAgentStatus objects	状态列表

表 4-65 ProtectableAgentStatus

参数	参数类型	描述
code	Integer	agent无法连接的错误码
installed	Boolean	agent是否安装
is_old	Boolean	agent是否为老版本
message	String	agent无法连接的错误信息
resource_id	String	资源ID
version	String	agent版本号

请求示例

查询云服务器资源agent状态

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/agent/check
```

```
{  
    "agent_status" : [ {  
        "resource_id" : "6c0e29fc-5eaa-4f76-af9e-cc79895f70d7",  
        "resource_name" : "ecs-8f7b",  
        "resource_type" : "OS::Nova::Server"  
    } ]  
}
```

响应示例

状态码： 200

OK

```
{  
    "agent_status" : [ {  
        "resource_id" : "6c0e29fc-5eaa-4f76-af9e-cc79895f70d7",  
        "installed" : false  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询云服务器资源agent状态

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CheckAgentSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CheckAgentRequest request = new CheckAgentRequest();  
        ProtectableAgentReq body = new ProtectableAgentReq();
```

```
List<ProtectableAgentStatusResource> listbodyAgentStatus = new ArrayList<>();
listbodyAgentStatus.add(
    new ProtectableAgentStatusResource()
        .withResourceId("6c0e29fc-5eaa-4f76-af9e-cc79895f70d7")
        .withResourceName("ecs-8f7b")
        .withResourceType("OS::Nova::Server")
);
body.withAgentStatus(listbodyAgentStatus);
request.withBody(body);
try {
    CheckAgentResponse response = client.checkAgent(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

查询云服务器资源agent状态

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CheckAgentRequest()
        listAgentStatusbody = [
            ProtectableAgentStatusResource(
                resource_id="6c0e29fc-5eaa-4f76-af9e-cc79895f70d7",
                resource_name="ecs-8f7b",
                resource_type="OS::Nova::Server"
            )
        ]
        request.body = ProtectableAgentReq(
            agent_status=listAgentStatusbody
        )
        response = client.check_agent(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

Go

查询云服务器资源agent状态

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CheckAgentRequest{}
    resourceNameAgentStatus:= "ecs-8f7b"
    var listAgentStatusbody = []model.ProtectableAgentStatusResource{
        {
            ResourceId: "6c0e29fc-5eaa-4f76-af9e-cc79895f70d7",
            ResourceName: &resourceNameAgentStatus,
            ResourceType: "OS::Nova::Server",
        },
    }
    request.Body = &model.ProtectableAgentReq{
        AgentStatus: listAgentStatusbody,
    }
    response, err := client.CheckAgent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.2.3 查询指定可保护资源

功能介绍

根据ID查询可保护性资源

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/protectables/{protectable_type}/instances/{instance_id}

表 4-66 路径参数

参数	是否必选	参数类型	描述
instance_id	是	String	资源ID
project_id	是	String	项目ID
protectable_type	是	String	可保护性类型 枚举值： • server • disk

请求参数

无

响应参数

状态码： 200

表 4-67 响应 Body 参数

参数	参数类型	描述
instance	Protectables Resp object	可保护资源

表 4-68 ProtectablesResp

参数	参数类型	描述
children	Array of objects	子资源
detail	Object	资源详情
id	String	id
name	String	名称
protectable	ProtectableR esult object	是否可备份
size	Integer	大小, 单位GB
status	String	资源状态 枚举值: • active • deleted • error
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-69 ProtectableResult

参数	参数类型	描述
code	String	不支持备份的错误码
reason	String	不支持备份的原因
result	Boolean	是否可备份
vault	VaultGet object	绑定的存储库
message	String	资源不可备份的原因信息, 当资源可保护性检验失败时才有该字段。

表 4-70 VaultGet

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’ 。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80

参数	参数类型	描述
sys_lock_source_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值: <ul style="list-style-type: none">• SMB• "
updated_at	String	更新时间,例如:"2020-02-05T10:38:34.209782"
version	String	版本

表 4-71 Billing

参数	参数类型	描述
allocated	Integer	已分配容量, 单位GB
charging_mode	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal

参数	参数类型	描述
status	String	存储库状态 枚举值： <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-72 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小

参数	参数类型	描述
backup_count	Integer	副本数量

表 4-73 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-74 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可 选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-75 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度：0 最大长度：5 数组长度：0 - 5

表 4-76 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

查询指定可保护资源。

```
GET https://{endpoint}/v3/{project_id}/protectables/{protectable_type}/instances/{instance_id}
```

响应示例

状态码： 200

OK

```
{  
  "instance": {  
    "status": "ACTIVE",  
    "children": [ {  
      "status": "in-use",  
      "name": "volume-f926-0001",  
      "detail": {  
        "attachments": [ {  
          "server_id": "55ecd8b8-1457-4a2a-a9df-53756a690082",  
          "attachment_id": "3b32cb75-bd96-493b-8fb2-13f063c876d5",  
          "volume_id": "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",  
          "attached_at": "2019-05-08T06:56:25.892+00:00",  
          "device": "/dev/vdb",  
          "id": "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99"  
        } ],  
        "links": [ {  
          "href": "https://volume.region1.com/v2/3aec9116a6a5487ca14987121d2818fe/volumes/8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99",  
          "rel": "self"  
        } ]  
      } ]  
  } }  
}
```

```
"href" : "https://volume.region1.com/3aec9116a6a5487ca14987121d2818fe/volumes/8d3d519fd409-4be3-8a1f-9f7a5ecb7a99",
    "rel" : "bookmark"
},
"availability_zone" : "AZ1",
"os-vol-host-attr:host" : "AZ1.xxxxxx#FusionStorage",
"encrypted" : false,
"updated_at" : "2019-05-08T06:56:26.207+00:00",
"replication_status" : "disabled",
"id" : "8d3d519fd409-4be3-8a1f-9f7a5ecb7a99",
"size" : 10,
"user_id" : "f3d66502790d4ddaaef5056059ed8f4",
"os-vol-tenant-attr:tenant_id" : "3aec9116a6a5487ca14987121d2818fe",
"metadata" : {
    "policy" : "cf3274fc-d23e-4c12-8be7-1e41b3c2faea",
    "readonly" : "False",
    "attached_mode" : "rw"
},
"status" : "in-use",
"multiattach" : false,
"name" : "volume-f926-0001",
"bootable" : "false",
"created_at" : "2019-03-26T09:30:46.784+00:00",
"volume_type" : "SATA",
"shareable" : false
},
"children" : [ ],
"type" : "OS::Cinder::Volume",
"id" : "8d3d519fd409-4be3-8a1f-9f7a5ecb7a99",
"size" : 10
}, {
    "status" : "in-use",
    "name" : "no_delete_for_cbr_interface-volume-0000",
    "detail" : {
        "attachments" : [ {
            "server_id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",
            "attachment_id" : "e16631fa-0522-4ac8-8f78-19db435caf68",
            "volume_id" : "13eb535f-8635-4faf-9f08-584d21e78f0a",
            "attached_at" : "2019-05-08T06:55:26.564+00:00",
            "device" : "/dev/vda",
            "id" : "13eb535f-8635-4faf-9f08-584d21e78f0a"
        }],
        "links" : [ {
            "href" : "https://volume.region1.com/v2/3aec9116a6a5487ca14987121d2818fe/volumes/13eb535f-8635-4faf-9f08-584d21e78f0a",
            "rel" : "self"
        },
        {
            "href" : "https://volume.region1.com/3aec9116a6a5487ca14987121d2818fe/volumes/13eb535f-8635-4faf-9f08-584d21e78f0a",
            "rel" : "bookmark"
        }
    ],
    "availability_zone" : "AZ1",
    "os-vol-host-attr:host" : "AZ1.xxxxxx#kvm_fusionstorage",
    "encrypted" : false,
    "updated_at" : "2019-05-08T08:33:07.855+00:00",
    "replication_status" : "disabled",
    "id" : "13eb535f-8635-4faf-9f08-584d21e78f0a",
    "size" : 40,
    "user_id" : "3204777a731c4ebea330350c04f41b92",
    "os-vol-tenant-attr:tenant_id" : "3aec9116a6a5487ca14987121d2818fe",
    "metadata" : {
        "policy" : "2ecbcffd-08ba-43f7-bf8d-95daba753464",
        "readonly" : "False",
        "attached_mode" : "rw"
    },
    "status" : "in-use",
    "volume_image_metadata" : {
        "container_format" : "bare",
        "min_ram" : 0,
```

```
        "_account_code" : "",  
        "_os_bit" : 64,  
        "size" : 2,  
        "_os_version" : "CentOS 7.2 64bit",  
        "_backup_id" : "",  
        "_description" : "",  
        "_support_diskintensive" : "true",  
        "disk_format" : "vhd",  
        "_isregistered" : "true",  
        "_image_size" : 3295645696,  
        "_platform" : "CentOS",  
        "_root_origin" : "file",  
        "_originalimagename" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",  
        "_quick_start" : "true",  
        "image_name" : "CentOS7.2-cloudinit",  
        "image_id" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",  
        "_os_type" : "Linux",  
        "_image_location" : "172.100.45.130:443:pcsimsf8bd00bf6c304d2e8cf4e2beb0aed702:703eb90a-a65f-4c72-b4c1-f070a55d068b",  
        "min_disk" : 40,  
        "_data_origin" : "file,jingxiang:CentOS-7.2-64bit.zvhed2",  
        "_support_kvm" : "true",  
        "virtual_env_type" : "FusionCompute",  
        "_image_source_type" : "uds",  
        "_support_xen" : "true",  
        "checksum" : "99914b932bd37a50b983c5e7c90ae93b",  
        "_imagetype" : "gold",  
        "_productcode" : ""  
    },  
    "multiattach" : false,  
    "name" : "no_delete_for_cbr_interface-volume-0000",  
    "bootable" : "true",  
    "created_at" : "2019-05-08T06:54:58.478+00:00",  
    "volume_type" : "SATA",  
    "shareable" : false  
},  
"children" : [ ],  
"type" : "OS::Cinder::Volume",  
"id" : "13eb535f-8635-4faf-9f08-584d21e78f0a",  
"size" : 40  
} ],  
"protectable" : {  
    "reason" : "",  
    "code" : "",  
    "result" : true  
},  
"name" : "no_delete_for_cbr_interface",  
"detail" : {  
    "addresses" : {  
        "94b88790-c906-4450-b7c2-1c2b1ea639bf" : [ {  
            "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:9b:9d:af",  
            "version" : 4,  
            "addr" : "192.168.1.44",  
            "OS-EXT-IPS:type" : "fixed"  
        }, {  
            "OS-EXT-IPS-MAC:mac_addr" : "fa:16:3e:9b:9d:af",  
            "version" : 4,  
            "addr" : "100.78.0.199",  
            "OS-EXT-IPS:type" : "floating"  
        } ]  
    },  
    "links" : [ {  
        "href" : "https://compute.region1.com/v2.1/3aec9116a6a5487ca14987121d2818fe/servers/55ecd8b8-1457-4a2a-a9df-53756a690082",  
        "rel" : "self"  
    }, {  
        "href" : "https://compute.region1.com/3aec9116a6a5487ca14987121d2818fe/servers/55ecd8b8-1457-4a2a-a9df-53756a690082",  
        "rel" : "bookmark"  
    } ]  
},  
"links" : [ {  
    "rel" : "bookmark",  
    "href" : "https://compute.region1.com/3aec9116a6a5487ca14987121d2818fe/servers/55ecd8b8-1457-4a2a-a9df-53756a690082"  
}, {  
    "rel" : "self",  
    "href" : "https://compute.region1.com/v2.1/3aec9116a6a5487ca14987121d2818fe/servers/55ecd8b8-1457-4a2a-a9df-53756a690082"  
} ]
```

```
        },
        "image" : "",
        "OS-EXT-STS:vm_state" : "active",
        "OS-EXT-SRV-ATTR:instance_name" : "instance-0004456e",
        "OS-SRV-USG:launched_at" : "2019-05-08T06:55:38.000+00:00",
        "flavor" : {
            "id" : "s3.small.1",
            "links" : [ {
                "href" : "https://compute.region1.com/3aec9116a6a5487ca14987121d2818fe/flavors/s3.small.1",
                "rel" : "bookmark"
            } ]
        },
        "id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",
        "security_groups" : [ {
            "name" : "default"
        } ],
        "user_id" : "f3d66502790d4ddaadef5056059ed8f4",
        "OS-DCF:diskConfig" : "MANUAL",
        "accessIPv4" : "",
        "accessIPv6" : "",
        "progress" : 0,
        "OS-EXT-STS:power_state" : 1,
        "OS-EXT-AZ:availability_zone" : "AZ1",
        "config_drive" : "",
        "status" : "ACTIVE",
        "updated" : "2019-05-08T06:57:02.000+00:00",
        "hostId" : "179f22cafe6b951921bd6deb3b13a978df620427fb37bbb7c619de98",
        "OS-EXT-SRV-ATTR:host" : "AZ1.xxxxxx",
        "OS-EXT-SRV-ATTR:hypervisor_hostname" : "nova003@36",
        "name" : "no_delete_for_cbr_interface",
        "created" : "2019-05-08T06:55:22.000+00:00",
        "tenant_id" : "3aec9116a6a5487ca14987121d2818fe",
        "os-extended-volumes:volumes_attached" : [ {
            "id" : "8d3d519f-d409-4be3-8a1f-9f7a5ecb7a99"
        }, {
            "id" : "13eb535f-8635-4faf-9f08-584d21e78f0a"
        } ],
        "metadata" : {
            "metering.image_id" : "703eb90a-a65f-4c72-b4c1-f070a55d068b",
            "metering.imagetype" : "gold",
            "metering.resourcespecocode" : "s3.small.1.linux",
            "metering.cloudServiceType" : "sys.service.type.ec2",
            "virtual_env_type" : "FusionCompute",
            "image_name" : "CentOS7.2-cloudinit",
            "metering.resourcetype" : 1,
            "vpc_id" : "94b88790-c906-4450-b7c2-1c2b1ea639bf",
            "os_bit" : 64,
            "cascaded.instance_extrainfo" : "",
            "os_type" : "Linux",
            "charging_mode" : 0
        }
    },
    "type" : "OS::Nova::Server",
    "id" : "55ecd8b8-1457-4a2a-a9df-53756a690082",
    "size" : 50
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowProtectableSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowProtectableRequest request = new ShowProtectableRequest();
        try {
            ShowProtectableResponse response = client.showProtectable(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.v1 import *

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

    credentials = BasicCredentials(ak, sk) \\\
        client = CbrClient.new_builder() \\
            .with_credentials(credentials) \\
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \\
            .build()
```

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

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ShowProtectableRequest{}  
    response, err := client.ShowProtectable(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.2.4 查询复制能力

功能介绍

查询本区域的复制能力

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/replication-capabilities

表 4-77 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-78 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-79 响应 Body 参数

参数	参数类型	描述
regions	Array of ProtectableRegion ReplicationCapabilitiesResponseRegion objects	支持复制的区域列表

表 4-80 ProtectableReplicationCapabilitiesRespRegion

参数	参数类型	描述
name	String	云服务所在的区域
replication_destinations	Array of strings	支持复制的目标区域列表

请求示例

```
GET https://{{endpoint}}/v3/{{project_id}}/replication-capabilities
```

响应示例

状态码： 200

OK

```
{  
    "regions": [ {  
        "replication_destinations": [ "cn-shenzhen-1", "cn-hk1" ],  
        "name": "southchina"  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowReplicationCapabilitiesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
    }  
}
```

```
CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ShowReplicationCapabilitiesRequest request = new ShowReplicationCapabilitiesRequest();
try {
    ShowReplicationCapabilitiesResponse response = client.showReplicationCapabilities(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3 备份

4.3.1 同步备份

功能介绍

同步线下混合云VMware备份副本

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/backups/sync

表 4-81 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-82 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-83 请求 Body 参数

参数	是否必选	参数类型	描述
sync	是	Array of BackupSync objects	待同步备份副本列表 数组长度：1 - 50

表 4-84 BackupSync

参数	是否必选	参数类型	描述
backup_id	是	String	备份副本ID
backup_name	是	String	备份名称 最小长度：1 最大长度：64
bucket_name	是	String	桶名
image_path	是	String	备份链在存储单元上的路径
resource_id	是	String	备份对象ID
resource_name	是	String	备份对象名称

参数	是否必选	参数类型	描述
resource_type	是	String	备份对象资源类型 缺省值: OS::Native::Server
created_at	是	Integer	备份时间戳, 例如1548898428

响应参数

状态码: 200

表 4-85 响应 Body 参数

参数	参数类型	描述
sync	Array of BackupSyncRespBody objects	同步备份副本接口的返回信息

表 4-86 BackupSyncRespBody

参数	参数类型	描述
backup_id	String	备份副本ID
operation_log_id	String	同步任务ID

请求示例

同步线下混合云VMware备份副本。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/backups/sync

{
  "sync": [ {
    "backup_id": "1f9e6b47-7655-4a19-b563-21b4ef023025",
    "resource_id": "96f7512d-239d-3f31-80cf-d1ce041df9a6",
    "resource_name": "resource_name",
    "resource_type": "OS::Native::Server",
    "bucket_name": "1555230361623e4ce4700dab84b7d8169a5eaec05227d",
    "image_path": "6abec15e-7baf-4d79-b5a9-6b98a2da7020/cfe0295c-5939-4b31-96cd-161b4725e7ad/cfe0295c-5939-4b31-96cd-161b4725e7ad",
    "created_at": 1553587260,
    "backup_name": "backup_name"
  } ]
}
```

响应示例

状态码: 200

OK

```
{  
    "sync": [  
        {  
            "backup_id": "1f9e6b47-7655-4a19-b563-21b4ef023025",  
            "operation_log_id": "154f0a9d-ba53-4e91-a3e2-a918710a9e0d"  
        }  
    ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

同步线下混合云VMware备份副本。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ImportBackupSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ImportBackupRequest request = new ImportBackupRequest();  
        BackupSyncReq body = new BackupSyncReq();  
        List<BackupSync> listbodySync = new ArrayList<>();  
        listbodySync.add(  
            new BackupSync()  
                .withBackupId("1f9e6b47-7655-4a19-b563-21b4ef023025")  
                .withBackupName("backup_name")  
                .withBucketName("1555230361623e4ce4700dab84b7d8169a5eaec05227d")  
                .withImagePath("6abec15e-7ba5-4d79-b5a9-6b98a2da7020/  
cfe0295c-5939-4b31-96cd-161b4725e7ad/cfe0295c-5939-4b31-96cd-161b4725e7ad")  
                .withResourceId("96f7512d-239d-3f31-80cf-d1ce041df9a6")  
                .withResourceName("resource_name")  
                .withResourceType("OS::Native::Server")  
                .withCreatedAt(1553587260)  
        );  
        body.withSync(listbodySync);  
        request.withBody(body);  
    }  
}
```

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

Python

同步线下混合云VMware备份副本。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ImportBackupRequest()
        listSyncbody = [
            BackupSync(
                backup_id="1f9e6b47-7655-4a19-b563-21b4ef023025",
                backup_name="backup_name",
                bucket_name="1555230361623e4ce4700dab84b7d8169a5eaec05227d",
                image_path="6abec15e-7baf-4d79-b5a9-6b98a2da7020/cfe0295c-5939-4b31-96cd-161b4725e7ad",
                resource_id="96f7512d-239d-3f31-80cf-d1ce041df9a6",
                resource_name="resource_name",
                resource_type="OS::Native::Server",
                created_at=1553587260
            )
        ]
        request.body = BackupSyncReq(
            sync=listSyncbody
        )
        response = client.import_backup(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

同步线下混合云VMware备份副本。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.ImportBackupRequest{}
    var listSyncbody = []model.BackupSync{
        {
            BackupId: "1f9e6b47-7655-4a19-b563-21b4ef023025",
            BackupName: "backup_name",
            BucketName: "1555230361623e4ce4700dab84b7d8169a5eaec05227d",
            ImagePath: "6abec15e-7baf-4d79-b5a9-6b98a2da7020/cfe0295c-5939-4b31-96cd-161b4725e7ad/
cfe0295c-5939-4b31-96cd-161b4725e7ad",
            ResourceId: "96f7512d-239d-3f31-80cf-d1ce041df9a6",
            ResourceName: "resource_name",
            ResourceType: "OS::Native::Server",
            CreatedAt: int32(1553587260),
        },
    }
    request.Body = &model.BackupSyncReq{
        Sync: listSyncbody,
    }
    response, err := client.ImportBackup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3.2 查询指定备份

功能介绍

根据指定id查询单个副本。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/backups/{backup_id}

表 4-87 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份ID
project_id	是	String	项目ID

请求参数

表 4-88 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-89 响应 Body 参数

参数	参数类型	描述
backup	BackupResp object	备份副本

表 4-90 BackupResp

参数	参数类型	描述
checkpoint_id	String	还原点ID
created_at	String	创建时间, 例如:"2020-02-05T10:38:34.209782"
description	String	备份描述
expired_at	String	过期时间, 例如:"2020-02-05T10:38:34.209782"
extend_info	BackupExtentionInfo object	扩展信息
id	String	备份ID
image_type	String	备份类型。取值为backup和replication。
name	String	备份名称
parent_id	String	父备份ID
project_id	String	项目ID
protected_at	String	备份时间
resource_az	String	资源可用区
resource_id	String	资源ID
resource_name	String	资源名称
resource_size	Integer	资源大小, 单位为GB
resource_type	String	资源类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

参数	参数类型	描述
status	String	备份状态 枚举值： <ul style="list-style-type: none">● available● protecting● deleting● restoring● error● waiting_protect● waiting_delete● waiting_restore
updated_at	String	更新时间，例如："2020-02-05T10:38:34.209782"
vault_id	String	存储库ID
replication_records	Array of ReplicationRecordGet objects	复制记录
enterprise_project_id	String	企业项目id,默认为‘0’。
provider_id	String	备份提供商ID，用于区分备份对象。当前取值包含： 0daac4c5-6707-4851-97ba-169e36266b66，该值代表备份对象为云服务器。 d1603440-187d-4516-af25-121250c7cc97，该值代表备份对象为云硬盘。 3f3c3220-245c-4805-b811-758870015881，该值代表备份对象为SFS Turbo。 a13639de-00be-4e94-af30-26912d75e4a2，该值代表备份对象为混合云VMware备份。
children	Array of BackupResp objects	子副本列表

表 4-91 BackupExtendInfo

参数	参数类型	描述
auto_trigger	Boolean	是否是自动生成的备份副本
bootable	Boolean	是否系统盘备份
incremental	Boolean	是否是增备

参数	参数类型	描述
snapshot_id	String	卷备份副本的快照id
support_lld	Boolean	是否支持lazyloading快速恢复
supported_restore_mode	String	备份支持恢复的方式，当前取值包含na,snapshot和backup。如果该字段取值为snapshot，代表备份此时已经支持创建整机镜像；如果该字段取值为backup，备份支持通过云服务器上硬盘的备份进行恢复；如果该字段取值为na，备份不支持直接恢复，只支持备份创建新资源等操作。 缺省值： na 枚举值： <ul style="list-style-type: none">• na• backup• snapshot
os_images_data	Array of ImageData objects	备份注册镜像ID列表
contain_system_disk	Boolean	整机备份是否包含系统盘
encrypted	Boolean	是否加密
system_disk	Boolean	是否是系统盘
is_multi_az	Boolean	备份类型是否为多AZ 缺省值： false

表 4-92 ImageData

参数	参数类型	描述
image_id	String	镜像ID

表 4-93 ReplicationRecordGet

参数	参数类型	描述
created_at	String	复制的开始时间
destination_backup_id	String	复制的目的备份ID
destination_checkpoint_id	String	复制的目的备份记录ID

参数	参数类型	描述
destination_project_id	String	复制的目标项目ID
destination_region	String	复制的目标区域
destination_vault_id	String	目标存储库ID
extra_info	ReplicationRecordsExtraInfo object	复制附加信息
id	String	复制记录ID
source_backup_id	String	复制的源备份ID
source_checkpoint_id	String	复制的源备份记录ID
source_project_id	String	复制的源项目ID
source_region	String	复制的源区域
status	String	复制的状态 枚举值： <ul style="list-style-type: none">• replicating• success• fail• skip• waiting_replicate
vault_id	String	备份所在的存储库ID

表 4-94 ReplicationRecordsExtraInfo

参数	参数类型	描述
progress	Integer	复制进度
fail_code	String	失败错误码，成功时为空
fail_reason	String	错误原因
auto_trigger	Boolean	是否为自动调度复制
destination_vault_id	String	目标端的存储库id

请求示例

查询单个备份副本。

```
GET https://{{endpoint}}/v3/{{project_id}}/backups/{{backup_id}}
```

响应示例

状态码： 200

OK

```
{  
    "backup": {  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "checkpoint_id": "8b0851a8-adf3-4f4c-a914-dead08bf9664",  
        "enterprise_project_id": 0,  
        "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",  
        "id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",  
        "resource_az": "az1.dc1",  
        "image_type": "backup",  
        "resource_id": "94eba8b2-acc9-4d82-badc-127144cc5526",  
        "resource_size": 40,  
        "children": [ {  
            "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
            "checkpoint_id": "8b0851a8-adf3-4f4c-a914-dead08bf9664",  
            "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",  
            "id": "5d822633-2bbf-4af8-a16e-5ab1c7705235",  
            "image_type": "backup",  
            "resource_id": "eccbcfdd-f843-4bbb-b2c0-a5ce861f9376",  
            "resource_size": 40,  
            "children": [ ],  
            "parent_id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",  
            "extend_info": {  
                "auto_trigger": true,  
                "snapshot_id": "5230a977-1a94-4092-8edd-519303a44cda",  
                "bootable": true,  
                "encrypted": true  
            },  
            "project_id": "4229d7a45436489f8c3dc2b1d35d4987",  
            "status": "available",  
            "resource_name": "ecs-1f0f-0002",  
            "name": "autobk_a843_ecs-1f0f-0002",  
            "created_at": "2019-05-10T07:59:59.451+00:00",  
            "resource_type": "OS::Cinder::Volume"  
        } ],  
        "extend_info": {  
            "auto_trigger": true,  
            "supported_restore_mode": "backup",  
            "contain_system_disk": true,  
            "support_lld": true  
        },  
        "project_id": "4229d7a45436489f8c3dc2b1d35d4987",  
        "status": "available",  
        "resource_name": "ecs-1f0f-0002",  
        "description": "backup_description",  
        "name": "backup_name",  
        "created_at": "2019-05-10T07:59:12.085+00:00",  
        "resource_type": "OS::Nova::Server"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowBackupSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowBackupRequest request = new ShowBackupRequest();
        try {
            ShowBackupResponse response = client.showBackup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \n\nclient = CbrClient.new_builder() \n    .with_credentials(credentials) \n    .with_region(CbrRegion.value_of("<YOUR REGION>")) \n    .build()\n\ntry:\n    request = ShowBackupRequest()\n    response = client.show_backup(request)\n    print(response)\nexcept exceptions.ClientRequestException as e:\n    print(e.status_code)\n    print(e.request_id)\n    print(e.error_code)\n    print(e.error_msg)
```

Go

```
package main\n\nimport (\n    "fmt"\n    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"\n    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"\n    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"\n    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/region"\n)\n\nfunc main() {\n    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security\n    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment\n    // variables and decrypted during use to ensure security.\n    // In this example, AK and SK are stored in environment variables for authentication. Before running this\n    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment\n    ak := os.Getenv("CLOUD_SDK_AK")\n    sk := os.Getenv("CLOUD_SDK_SK")\n\n    auth := basic.NewCredentialsBuilder().\n        WithAk(ak).\n        WithSk(sk).\n        Build()\n\n    client := cbr.NewCbrClient(\n        cbr.CbrClientBuilder().\n            WithRegion(region.ValueOf("<YOUR REGION>")).\n            WithCredential(auth).\n            Build())\n\n    request := &model.ShowBackupRequest{}\n    response, err := client.ShowBackup(request)\n    if err == nil {\n        fmt.Printf("%+v\n", response)\n    } else {\n        fmt.Println(err)\n    }\n}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3.3 查询所有备份

功能介绍

查询所有副本

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/backups

表 4-95 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-96 Query 参数

参数	是否必选	参数类型	描述
checkpoint_id	否	String	还原点ID
dec	否	Boolean	专属云（专属云场景使用，非专属云场景不生效）
end_time	否	String	备份产生时间范围的结束时间，格式为%YYYY-%mm-%ddT%HH:%MM:%SSZ，例如2018-02-01T12:00:00Z
enterprise_project_id	否	String	企业项目id或all_granted_eps，all_granted_eps表示查询用户有权限的所有企业项目id
image_type	否	String	备份类型。取值为backup和replication。

参数	是否必选	参数类型	描述
incremental	否	Boolean	是否是增备 缺省值: false
limit	否	Integer	每页显示的条目数量, 正整数
marker	否	String	上一次查询最后一条的id
member_status	否	String	共享状态 枚举值: <ul style="list-style-type: none">• pending• accepted• rejected
name	否	String	名称
offset	否	Integer	偏移值, 正整数
own_type	否	String	持有类型, 私有的private/共享的shared/全部all_granted, 默认只查询private。 缺省值: private 枚举值: <ul style="list-style-type: none">• all_granted• private• shared
parent_id	否	String	父备份ID
resource_az	否	String	支持按az来过滤
resource_id	否	String	资源ID
resource_name	否	String	资源名称
resource_type	否	String	资源类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
show_replication	否	Boolean	是否返回复制记录 缺省值: false

参数	是否必选	参数类型	描述
sort	否	String	sort的内容为一组由逗号分隔的属性及可选排序方向组成，形如<key1>[:<direction>],<key2>[:<direction>]，其中direction的取值为asc (升序) 或 desc (降序)，如没有传入direction参数，默认为降序，sort内容的长度限制为255个字符。key取值范围：[created_at, updated_at, name, status, protected_at, id]
start_time	否	String	备份产生时间范围的开始时间，格式为%YYYY-%mm-%ddT%HH:%MM:%SSZ，例如2018-02-01T12:00:00Z
status	否	String	状态。调用API时，支持通过传多个status值进行过滤。例如：status=available&status=error枚举值： <ul style="list-style-type: none">• available• protecting• deleting• restoring• error• waiting_protect• waiting_delete• waiting_restore
used_percent	否	String	根据存储库使用率过滤备份，取值范围 [1, 100]，含1和100。例如，used_percent=80，表示筛选所属存储库使用率大于等于80%的所有备份。
vault_id	否	String	存储库ID

请求参数

表 4-97 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-98 响应 Body 参数

参数	参数类型	描述
backups	Array of BackupResp objects	备份列表
count	Integer	备份个数
offset	Integer	偏移量，表示从此偏移量开始查询 最小值： 0 缺省值： 0
limit	Integer	每页显示的条目数量 最小值： 1 最大值： 1000 缺省值： 1000

表 4-99 BackupResp

参数	参数类型	描述
checkpoint_id	String	还原点ID
created_at	String	创建时间，例如："2020-02-05T10:38:34.209782"
description	String	备份描述
expired_at	String	过期时间，例如："2020-02-05T10:38:34.209782"
extend_info	BackupExtende dInfo object	扩展信息
id	String	备份ID

参数	参数类型	描述
image_type	String	备份类型。取值为backup和replication。
name	String	备份名称
parent_id	String	父备份ID
project_id	String	项目ID
protected_at	String	备份时间
resource_az	String	资源可用区
resource_id	String	资源ID
resource_name	String	资源名称
resource_size	Integer	资源大小，单位为GB
resource_type	String	资源类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
status	String	备份状态 枚举值: <ul style="list-style-type: none">● available● protecting● deleting● restoring● error● waiting_protect● waiting_delete● waiting_restore
updated_at	String	更新时间，例如:"2020-02-05T10:38:34.209782"
vault_id	String	存储库ID
replication_records	Array of ReplicationRecordGet objects	复制记录
enterprise_project_id	String	企业项目id,默认为‘0’。

参数	参数类型	描述
provider_id	String	备份提供商ID，用于区分备份对象。当前取值包含： 0daac4c5-6707-4851-97ba-169e36266b66，该值代表备份对象为云服务器。 d1603440-187d-4516-af25-121250c7cc97，该值代表备份对象为云硬盘。 3f3c3220-245c-4805-b811-758870015881，该值代表备份对象为SFS Turbo。 a13639de-00be-4e94-af30-26912d75e4a2，该值代表备份对象为混合云VMware备份。
children	Array of BackupResp objects	子副本列表

表 4-100 BackupExtendInfo

参数	参数类型	描述
auto_trigger	Boolean	是否是自动生成的备份副本
bootable	Boolean	是否系统盘备份
incremental	Boolean	是否是增备
snapshot_id	String	卷备份副本的快照id
support_lld	Boolean	是否支持lazyloading快速恢复
supported_restore_mode	String	备份支持恢复的方式，当前取值包含na,snapshot和backup。如果该字段取值为snapshot，代表备份此时已经支持创建整机镜像；如果该字段取值为backup，备份支持通过云服务器上硬盘的备份进行恢复；如果该字段取值为na，备份不支持直接恢复，只支持备份创建新资源等操作。 缺省值：na 枚举值： <ul style="list-style-type: none">• na• backup• snapshot
os_images_data	Array of ImageData objects	备份注册镜像ID列表
contain_system_disk	Boolean	整机备份是否包含系统盘
encrypted	Boolean	是否加密

参数	参数类型	描述
system_disk	Boolean	是否是系统盘
is_multi_az	Boolean	备份类型是否为多AZ 缺省值: false

表 4-101 ImageData

参数	参数类型	描述
image_id	String	镜像ID

表 4-102 ReplicationRecordGet

参数	参数类型	描述
created_at	String	复制的开始时间
destination_backup_id	String	复制的目的备份ID
destination_checkpoint_id	String	复制的目的备份记录ID
destination_project_id	String	复制的目标项目ID
destination_region	String	复制的目标区域
destination_vault_id	String	目标存储库ID
extra_info	ReplicationRecordsExtraInfo object	复制附加信息
id	String	复制记录ID
source_backup_id	String	复制的源备份ID
source_checkpoint_id	String	复制的源备份记录ID
source_project_id	String	复制的源项目ID
source_region	String	复制的源区域

参数	参数类型	描述
status	String	复制的状态 枚举值： <ul style="list-style-type: none">• replicating• success• fail• skip• waiting_replicate
vault_id	String	备份所在的存储库ID

表 4-103 ReplicationRecordsExtraInfo

参数	参数类型	描述
progress	Integer	复制进度
fail_code	String	失败错误码，成功时为空
fail_reason	String	错误原因
auto_trigger	Boolean	是否为自动调度复制
destination_vault_id	String	目标端的存储库id

请求示例

查询所有备份副本。

```
GET https://{endpoint}/v3/{project_id}/backups
```

响应示例

状态码： 200

OK

```
{  
    "count" : 2,  
    "backups" : [ {  
        "provider_id" : "0daac4c5-6707-4851-97ba-169e36266b66",  
        "checkpoint_id" : "1fcfed58b-2a31-4851-bcbb-96216f83ce99",  
        "updated_at" : "2020-02-21T07:07:25.114+00:00",  
        "vault_id" : "cca85ea5-00a4-418d-9222-bd83985bc515",  
        "id" : "b1c4af9-e7a6-4888-9010-c2bac3aa7910",  
        "resource_az" : "br-iaas-odin1a",  
        "image_type" : "backup",  
        "resource_id" : "1a503932-ee8f-4dd5-8248-8dfb57e584c5",  
        "resource_size" : 40,  
        "children" : [ ],  
        "extend_info" : {  
            "auto_trigger" : true,  
            "failover" : false  
        }  
    } ]  
}
```

```
        "supported_restore_mode" : "backup",
        "contain_system_disk" : true,
        "support_lld" : true,
        "system_disk" : false
    },
    "project_id" : "0605767b5780d5762fc5c0118072a564",
    "status" : "available",
    "resource_name" : "test001-02",
    "description" : "",
    "expired_at" : "2020-05-21T07:00:54.060+00:00",
    "name" : "autobk_b629",
    "created_at" : "2020-02-21T07:00:54.065+00:00",
    "resource_type" : "OS::Nova::Server"
}, {
    "provider_id" : "d1603440-187d-4516-af25-121250c7cc97",
    "checkpoint_id" : "f64c351f-769f-4c04-8806-fd90a59e9b12",
    "updated_at" : "2020-02-21T07:09:37.767+00:00",
    "vault_id" : "79bd9daa-884f-4f84-b8fe-235d58cd927d",
    "id" : "5606aab5-2dc2-4498-8144-dc848d099af5",
    "resource_az" : "br-iaas-odin1a",
    "image_type" : "backup",
    "resource_id" : "54f7ccbc-072f-4ec5-a7b7-b24dabdb4539",
    "resource_size" : 40,
    "children" : [ ],
    "extend_info" : {
        "auto_trigger" : true,
        "snapshot_id" : "e3def9a8-e4b4-4c12-b132-f4ba8ce9a34f",
        "bootable" : true,
        "support_lld" : true,
        "encrypted" : false,
        "system_disk" : false
    },
    "project_id" : "0605767b5780d5762fc5c0118072a564",
    "status" : "available",
    "resource_name" : "qsy_000",
    "description" : "",
    "expired_at" : "2020-03-22T07:00:34.878+00:00",
    "name" : "autobk_6809",
    "created_at" : "2020-02-21T07:00:34.882+00:00",
    "resource_type" : "OS::Cinder::Volume"
} ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ListBackupsSolution {

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

```

```
this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ListBackupsRequest request = new ListBackupsRequest();
request.withCheckpointId("<checkpoint_id>");
request.withDec(<dec>);
request.withEndTime("<end_time>");
request.withImageType(ListBackupsRequest.ImageTypeEnum.fromValue("<image_type>"));
request.withLimit(<limit>);
request.withMarker("<marker>");
request.withName("<name>");
request.withOffset(<offset>);
request.withResourceAz("<resource_az>");
request.withResourceId("<resource_id>");
request.withResourceName("<resource_name>");
request.withResourceType(ListBackupsRequest.ResourceTypeEnum.fromValue("<resource_type>"));
request.withSort("<sort>");
request.withStartTime("<start_time>");
request.withStatus(ListBackupsRequest.StatusEnum.fromValue("<status>"));
request.withVaultId("<vault_id>");
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withOwnType(ListBackupsRequest.OwnTypeEnum.fromValue("<own_type>"));
request.withMemberStatus(ListBackupsRequest.MemberStatusEnum.fromValue("<member_status>"));
request.withParentId("<parent_id>");
request.withUsedPercent("<used_percent>");
request.withShowReplication(<show_replication>);
request.withIncremental(<incremental>);
try {
    ListBackupsResponse response = client.listBackups(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
```

```
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = ListBackupsRequest()
    request.checkpoint_id = "<checkpoint_id>"
    request.dec = <Dec>
    request.end_time = "<end_time>"
    request.image_type = "<image_type>"
    request.limit = <limit>
    request.marker = "<marker>"
    request.name = "<name>"
    request.offset = <offset>
    request.resource_az = "<resource_az>"
    request.resource_id = "<resource_id>"
    request.resource_name = "<resource_name>"
    request.resource_type = "<resource_type>"
    request.sort = "<sort>"
    request.start_time = "<start_time>"
    request.status = "<status>"
    request.vault_id = "<vault_id>"
    request.enterprise_project_id = "<enterprise_project_id>"
    request.own_type = "<own_type>"
    request.member_status = "<member_status>"
    request.parent_id = "<parent_id>"
    request.used_percent = "<used_percent>"
    request.show_replication = <ShowReplication>
    request.incremental = <Incremental>
    response = client.list_backups(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region" "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
```

```
cbr.CbrClientBuilder().  
    WithRegion(region.ValueOf("<YOUR REGION>")).  
    WithCredential(auth).  
    Build()  
  
request := &model.ListBackupsRequest{}  
checkpointIdRequest:= "<checkpoint_id>"  
request.CheckpointId = &checkpointIdRequest  
decRequest:= <dec>  
request.Dec = &decRequest  
endTimeRequest:= "<end_time>"  
request.EndTime = &endTimeRequest  
imageTypeRequest:= model.GetListBackupsRequestImageTypeEnum().<IMAGE_TYPE>  
request.ImageType = &imageTypeRequest  
limitRequest:= int32(<limit>)  
request.Limit = &limitRequest  
markerRequest:= "<marker>"  
request.Marker = &markerRequest  
nameRequest:= "<name>"  
request.Name = &nameRequest  
offsetRequest:= int32(<offset>)  
request.Offset = &offsetRequest  
resourceAzRequest:= "<resource_az>"  
request.ResourceAz = &resourceAzRequest  
resourceIdRequest:= "<resource_id>"  
request.ResourceId = &resourceIdRequest  
resourceNameRequest:= "<resource_name>"  
request.ResourceName = &resourceNameRequest  
resourceTypeRequest:= model.GetListBackupsRequestResourceTypeEnum().<RESOURCE_TYPE>  
request.ResourceType = &resourceTypeRequest  
sortRequest:= "<sort>"  
request.Sort = &sortRequest  
startTimeRequest:= "<start_time>"  
request.StartTime = &startTimeRequest  
statusRequest:= model.GetListBackupsRequestStatusEnum().<STATUS>  
request.Status = &statusRequest  
vaultIdRequest:= "<vault_id>"  
request.VaultId = &vaultIdRequest  
enterpriseProjectIdRequest:= "<enterprise_project_id>"  
request.EnterpriseProjectId = &enterpriseProjectIdRequest  
ownTypeRequest:= model.GetListBackupsRequestOwnTypeEnum().<OWN_TYPE>  
request.OwnType = &ownTypeRequest  
memberStatusRequest:= model.GetListBackupsRequestMemberStatusEnum().<MEMBER_STATUS>  
request.MemberStatus = &memberStatusRequest  
parentIdRequest:= "<parent_id>"  
request.ParentId = &parentIdRequest  
usedPercentRequest:= "<used_percent>"  
request.UsedPercent = &usedPercentRequest  
showReplicationRequest:= <show_replication>  
request.ShowReplication = &showReplicationRequest  
incrementalRequest:= <incremental>  
request.Incremental = &incrementalRequest  
response, err := client.ListBackups(request)  
if err == nil {  
    fmt.Printf("%#+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3.4 删除备份

功能介绍

删除单个备份。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/backups/{backup_id}

表 4-104 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	指定删除的备份ID
project_id	是	String	项目ID

请求参数

表 4-105 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

无

请求示例

删除单个备份。

```
DELETE https://{endpoint}/v3/{project_id}/backups/{backup_id}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DeleteBackupSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteBackupRequest request = new DeleteBackupRequest();
        try {
            DeleteBackupResponse response = client.deleteBackup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBackupRequest{}
    response, err := client.DeleteBackup(request)
```

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

更多

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

状态码

状态码	描述
204	No Content

错误码

请参见[错误码](#)。

4.3.5 复制备份

功能介绍

跨区域复制备份。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/backups/{backup_id}/replicate

表 4-106 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	复制的备份ID
project_id	是	String	项目ID

请求参数

表 4-107 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-108 请求 Body 参数

参数	是否必选	参数类型	描述
replicate	是	BackupReplicateReqBody object	复制参数

表 4-109 BackupReplicateReqBody

参数	是否必选	参数类型	描述
description	否	String	复制的描述 最小长度: 0 最大长度: 255
destination_project_id	是	String	复制的目标项目ID
destination_region	是	String	复制的目标区域 最小长度: 0 最大长度: 255
destination_vault_id	是	String	复制的目标区域的存储库ID
enable_acceleration	否	Boolean	跨区域复制时，是否启用加速从而缩短复制的时间，如果不指定，默认不启用加速。 缺省值: false
name	否	String	复制名称 最小长度: 1 最大长度: 64

响应参数

状态码： 200

表 4-110 响应 Body 参数

参数	参数类型	描述
replication	BackupReplicateRespBody object	复制返回参数

表 4-111 BackupReplicateRespBody

参数	参数类型	描述
backup_id	String	待复制的备份ID
destination_project_id	String	复制的目标项目ID
destination_region	String	复制的目标区域
destination_vault_id	String	复制的目标区域存储库ID
project_id	String	执行复制的项目ID
provider_id	String	资源类型id
replication_record_id	String	复制记录ID
source_region	String	复制的源区域

请求示例

复制一个备份从源备份存储库到目标备份存储库。

```
POST https://{{endpoint}}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/backups/a5200268-74a5-4806-acc6-95793ab0228b/replicate

{
  "replicate": {
    "description": "backup_description",
    "destination_project_id": "68589cac08274b82b4e254268a3862d8",
    "destination_region": "region2",
    "destination_vault_id": "0ca3eb86-8800-46da-9c37-9d657a825274",
    "enable_acceleration": false,
    "name": "backup_name"
  }
}
```

响应示例

状态码： 200

OK

```
{  
    "replication": {  
        "destination_vault_id": "0ca3eb86-8800-46da-9c37-9d657a825274",  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "backup_id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",  
        "destination_project_id": "68589cac08274b82b4e254268a3862d8",  
        "destination_region": "region2",  
        "source_region": "region1",  
        "project_id": "4229d7a45436489f8c3dc2b1d35d4987",  
        "replication_record_id": "1579a71e-8d8d-41e6-85dc-d77f5ce8d91a"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

复制一个备份从源备份存储库到目标备份存储库。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class CopyBackupSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CopyBackupRequest request = new CopyBackupRequest();  
        BackupReplicateReq body = new BackupReplicateReq();  
        BackupReplicateReqBody replicatebody = new BackupReplicateReqBody();  
        replicatebody.withDescription("backup_description")  
            .withDestinationProjectId("68589cac08274b82b4e254268a3862d8")  
            .withDestinationRegion("region2")  
            .withDestinationVaultId("0ca3eb86-8800-46da-9c37-9d657a825274")  
            .withEnableAcceleration(false)  
            .withName("backup_name");
```

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

Python

复制一个备份从源备份存储库到目标备份存储库。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CopyBackupRequest()
        replicatebody = BackupReplicateReqBody(
            description="backup_description",
            destination_project_id="68589cac08274b82b4e254268a3862d8",
            destination_region="region2",
            destination_vault_id="0ca3eb86-8800-46da-9c37-9d657a825274",
            enable_acceleration=False,
            name="backup_name"
        )
        request.body = BackupReplicateReq(
            replicate=replicatebody
        )
        response = client.copy_backup(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

复制一个备份从源备份存储库到目标备份存储库。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CopyBackupRequest{
        DescriptionReplicate: "backup_description",
        EnableAccelerationReplicate: false,
        NameReplicate: "backup_name",
        Replicatebody: &model.BackupReplicateReqBody{
            Description: &descriptionReplicate,
            DestinationProjectId: "68589cac08274b82b4e254268a3862d8",
            DestinationRegion: "region2",
            DestinationVaultId: "0ca3eb86-8800-46da-9c37-9d657a825274",
            EnableAcceleration: &enableAccelerationReplicate,
            Name: &nameReplicate,
        },
        request.Body = &model.BackupReplicateReq{
            Replicate: replicatebody,
        }
    }
    response, err := client.CopyBackup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3.6 备份恢复

功能介绍

恢复备份数据

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/backups/{backup_id}/restore

表 4-112 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份id
project_id	是	String	项目id

请求参数

表 4-113 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-114 请求 Body 参数

参数	是否必选	参数类型	描述
restore	是	BackupRestore object	恢复请求参数体

表 4-115 BackupRestore

参数	是否必选	参数类型	描述
mappings	否	Array of BackupRestoreServerMapping objects	恢复的映射关系(整机恢复时必填, 卷恢复时可选但是不会用到填写的值)
power_on	否	Boolean	恢复后是否开始, 默认开机。 缺省值: true
server_id	否	String	恢复的目标虚拟机ID (整机恢复时必填)
volume_id	否	String	恢复的目标卷ID (卷恢复时必填)
resource_id	否	String	待恢复的目标资源ID
details	否	RestoreDetails object	恢复详情

表 4-116 BackupRestoreServerMapping

参数	是否必选	参数类型	描述
backup_id	是	String	卷备份ID, 可以通过控制台或者“查询指定备份”接口获取。
volume_id	是	String	待恢复目标卷ID

表 4-117 RestoreDetails

参数	是否必选	参数类型	描述
destination_path	是	String	目的路径 最小长度: 1 最大长度: 255

响应参数

无

请求示例

恢复指定备份数据到虚拟机。

```
POST https://{{endpoint}}/v3/{{f841e01fd2b14e7fa41b6ae7aa6b0594}}/backups/a5200268-74a5-4806-acc6-95793ab0228b/restore

{
  "restore": {
    "mappings": [ {
      "backup_id": "5d822633-2bbf-4af8-a16e-5ab1c7705235",
      "volume_id": "eccbcfdd-f843-4bbb-b2c0-a5ce861f9376"
    }],
    "power_on": true,
    "server_id": "94eba8b2-acc9-4d82-badc-127144cc5526"
  }
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

恢复指定备份数据到虚拟机。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class RestoreBackupSolution {

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

        CbrClient client = CbrClient.newBuilder()
```

```
.withCredential(auth)
.withRegion(CbrRegion.valueOf("<YOUR REGION>"))
.build();
RestoreBackupRequest request = new RestoreBackupRequest();
BackupRestoreReq body = new BackupRestoreReq();
List<BackupRestoreServerMapping> listRestoreMappings = new ArrayList<>();
listRestoreMappings.add(
    new BackupRestoreServerMapping()
        .withBackupId("5d822633-2bbf-4af8-a16e-5ab1c7705235")
        .withVolumeId("eccbcfdd-f843-4bbb-b2c0-a5ce861f9376")
);
BackupRestore restorebody = new BackupRestore();
restorebody.withMappings(listRestoreMappings)
    .withPowerOn(true)
    .withServerId("94eba8b2-acc9-4d82-badc-127144cc5526");
body.withRestore(restorebody);
request.withBody(body);
try {
    RestoreBackupResponse response = client.restoreBackup(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

恢复指定备份数据到虚拟机。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = RestoreBackupRequest()
        listMappingsRestore = [
            BackupRestoreServerMapping(
                backup_id="5d822633-2bbf-4af8-a16e-5ab1c7705235",
                volume_id="eccbcfdd-f843-4bbb-b2c0-a5ce861f9376"
            )
        ]
    
```

```
restorebody = BackupRestore(
    mappings=listMappingsRestore,
    power_on=True,
    server_id="94eba8b2-acc9-4d82-badc-127144cc5526"
)
request.body = BackupRestoreReq(
    restore=restorebody
)
response = client.restore_backup(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

恢复指定备份数据到虚拟机。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestoreBackupRequest{}
    var listMappingsRestore = []model.BackupRestoreServerMapping{
        {
            BackupId: "5d822633-2bbf-4af8-a16e-5ab1c7705235",
            VolumId: "eccbcfdd-f843-4bbb-b2c0-a5ce861f9376",
        },
    }
    powerOnRestore:= true
    serverIdRestore= "94eba8b2-acc9-4d82-badc-127144cc5526"
    restorebody := &model.BackupRestore{
        Mappings: &listMappingsRestore,
        PowerOn: &powerOnRestore,
        ServerId: &serverIdRestore,
    }
    request.Body = &model.BackupRestoreReq{
        Restore: restorebody,
    }
    response, err := client.RestoreBackup(request)
    if err == nil {
```

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

更多

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

状态码

状态码	描述
202	Accepted

错误码

请参见[错误码](#)。

4.3.7 查询备份元数据

功能介绍

查询备份时资源的元数据

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/backups/{backup_id}/metadata

表 4-118 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份ID
project_id	是	String	项目ID

请求参数

无

响应参数

状态码： 200

表 4-119 响应 Body 参数

参数	参数类型	描述
backup_id	String	备份ID
backups	String	云服务器备份信息
flavor	String	云服务器规格信息
floatingips	Array of strings	云服务器浮动IP信息
interface	String	云服务器接口信息
ports	Array of strings	云服务器端口信息
server	String	云服务器信息
volumes	Array of strings	云服务器卷信息

请求示例

```
GET https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/backups/metadata
```

响应示例

状态码： 200

OK

```
{  
    "backup_id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",  
    "backups": "{\"provider_id\": \"0daac4c5-6707-4851-97ba-169e36266b66\", \"checkpoint_id\": \"8b0851a8-adf3-4f4c-a914-dead08bf9664\", \"updated_at\": null, \"vault_id\": \"3b5816b5-f29c-4172-9d9a-76c719a659ce\", \"id\": \"6df2b54c-dd62-4059-a07c-1b8f24f2725d\", \"resource_az\": \"az1.dc1\", \"image_type\": \"backup\", \"resource_id\": \"94eba8b2-acc9-4d82-badc-127144cc5526\", \"resource_size\": 40, \"children\": [], \"parent_id\": null, \"extend_info\": {\"auto_trigger\": true, \"supported_restore_mode\": \"backup\", \"os_images_data\": null, \"contain_system_disk\": true, \"snapshot_id\": null, \"bootable\": null, \"progress\": null, \"support_lld\": true, \"app_consistency\": {\"app_consistency_error_code\": \"0\", \"app_consistency_status\": \"0\", \"app_consistency_error_message\": \"\"}, \"app_consistency\": \"0\"}, \"project_id\": \"4229d7a45436489f8c3dc2b1d35d4987\", \"status\": \"available\", \"resource_name\": \"ecs-1f0f-0002\", \"description\": \"backup_description\", \"expired_at\": null, \"replication_records\": [], \"name\": \"backup_name\", \"created_at\": \"2019-05-10T07:59:12.084695\", \"protected_at\": null, \"resource_type\": \"OS::Nova::Server\"},  
    "server": {"server": {"tags": ["key_01", "key_02"], "progress": 0, "links": [{"rel": "self", "href": "https://compute.region.dc1.domainname.com/v2.1/4229d7a45436489f8c3dc2b1d35d4987/servers/94eba8b2-acc9-4d82-badc-127144cc5526"}], "rel": "bookmark", "href": "https://compute.region.dc1.domainname.com/4229d7a45436489f8c3dc2b1d35d4987/servers/94eba8b2-acc9-4d82-badc-127144cc5526"}, "id": "94eba8b2-acc9-4d82-badc-127144cc5526", "name": "ecs-1f0f-0002", "status": "ACTIVE", "OS-EXT-STS:vm_state": "active", "OS-EXT-STS:power_state": "1", "OS-EXT-STS:task_state": null, "flavor": {"links": [{"rel": "bookmark", "href": "https://compute.region.dc1.domainname.com/4229d7a45436489f8c3dc2b1d35d4987/flavors/s2.small.1"}], "id": "s2.small.1"}, "accessIPv4": "", "accessIPv6": "", "image": "", "OS-EXT-SRV-ATTR:hostname": "ecs-1f0f-0002", "OS-EXT-SRV-ATTR:host": "az1.dc1", "tenant_id": "4229d7a45436489f8c3dc2b1d35d4987", "user_id": "38d65be2ecd840d19046e239e841a734", "OS-EXT-AZ:availability_zone": "az1.dc1", "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova001@248", "OS-EXT-SRV-ATTR:launch_index": 0, "OS-EXT-SRV-ATTR:user_data": "IyEvYmluL2Jhc2gKZWNoByAncm9vdDokNiQ2OUNsYVckUFNka1pZa2RicGFsTFJGRnNLejhJVGp4eUlSjNy a1M3UURaMlRmT0J3Nk9ndUvhRm8xdGhMRS43aWpvc3RFZTlZvKyuQjBGNm1UQURxZEhWbTkxDEnIHwg"},  
    "links": [{"rel": "self", "href": "https://compute.region.dc1.domainname.com/v2.1/4229d7a45436489f8c3dc2b1d35d4987/backups/metadata"}]  
}
```

```
Y2hwYXNzd2QgLWU7\";"OS-SRV-USG:launched_at":\\"2019-05-08T08:53:07.000000\","OS-SRV-USG:terminated_at":null,"config_drive":\\"\","created":\\"2019-05-08T08:52:22Z\","description":\\"\","key_name":null,"locked":false,"updated":\\"2019-05-08T08:53:12Z\","metadata":\{"charging_mode":\\"0\","image_name":\\"Public CentOS 7.6 64bit for Test\","metering.cloudServiceType":\\"hws.service.type.ec2\","metering.image_id":\\"79bee4ee-0025-4645-b004-23d2a66f6eec\","metering.imagetype":\\"gold\","metering.resourcespeccode":\\"s2.small.1.linux\","metering.resourcetype":\\"1\","os_bit":\\"64\","os_type":\\"Linux\","vpc_id":\\"df963ec6-561c-4c69-9787-0456a55f8fd0\","hostId":\\"bfced2732a5c5f974a1882616828d31d32c0d79964d26b1f3d8694e3\","OS-EXT-SRV-ATTR:kernel_id":\\"\\","OS-EXT-SRV-ATTR:ramdisk_id":\\"\\","OS-EXT-SRV-ATTR:root_device_name":\\"/dev/vda\","os-extended-volumes:volumes_attached":\[\{"id":\\"eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\","delete_on_termination":true\}],\\"OS-DCF:diskConfig":\\"MANUAL\","security_groups":\[\{"name":\\"sg-a924\"}\],\\"OS-EXT-SRV-ATTR:reservation_id":\\"r-y43be5p0\","addresses":\{\\"df963ec6-561c-4c69-9787-0456a55f8fd0\":\[\{"OS-EXT-IPS-MAC:mac_addr":\\"fa:16:3e:ef:d5:fb\","version":4,\\"addr":\\"192.168.1.128\","OS-EXT-IPS:type":\\"fixed\","OS-EXT-IPS-MAC:mac_addr":\\"fa:16:3e:ef:d5:fb\","version":4,\\"addr":\\"100.64.1.29\","OS-EXT-IPS:type":\\"floating\\"\}\]},\\"host_status":\\"UP\","OS-EXT-SRV-ATTR:instance_name":\\"instance-00260179\\"",\\"volumes":\[\{"volume":\\"attachments\":[\{"attached_at":\\"2019-05-08T08:52:48.810938\","attachment_id":\\"8cbc141a-9403-48e2-9323-6c05683e7e50\","device":\\"/dev/vda\","host_name":null,\\"id":\\"eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\","server_id":\\"94eba8b2-acc9-4d82-badc-127144cc5526\","volume_id":\\"eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\\"",\\"availability_zone":\\"az1.dc1\","backup_id":null,\\"bootable":true\,"consistencygroup_id":null,\\"created_at":\\"2019-05-08T08:52:26.824375\","dedicated_storage_id":\\"\\","dedicated_storage_name":null,\\"description":\\"\\","encrypted":false,\\"enterprise_project_id":\\"0\","id":\\"eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\","lifecycle":0,\\"links":\[\{"href":\\"https://evs.cn-north-1.myhuaweicloud.com/v2/4229d7a45436489f8c3dc2b1d35d4987/os-vendor-volumes/eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\","rel":\\"self\","href":\\"https://evs.cn-north-1.myhuaweicloud.com/4229d7a45436489f8c3dc2b1d35d4987/os-vendor-volumes/eccbcfdd-f843-4bbb-b2c0-a5ce861f9376\","rel":\\"bookmark\\"\},\\"metadata":\[\{"openstack_region_name":\\"az1.dc1\","system_volume_name":\\"ecs-1f0f-0002\","attached_mode":\\"rw\","readonly":\\"False\","migration_status":null,\\"multiattach":false,\\"name":\\"ecs-1f0f-0002\","os-vol-host-attr:host":\\"az1.dc1#2\","os-vol-mig-status-attr:migstat":null,\\"os-vol-mig-status-attr:name_id":null,\\"os-vol-tenant-attr:tenant_id":\\"4229d7a45436489f8c3dc2b1d35d4987\","os-volume-replication:extended_status":null,\\"plan_delete_at":null,\\"pre_deleted_at":null,\\"provider_location":\\"{\\"storage_type\": \\"FC_DSWARE\", \\"vol_name\": \\"eccbcfdd8434bbbb2c0a5ce861f9376\","offset":4,\\"ip":\\"192.144.44.120\","urn":\\"urn:sites:4D760908:Volumes:3975388\","uri":\\"/service/sites/4D760908/Volumes/3975388\","pool":\\"2\\"",\\"replication_status":\\"disabled\","restored_from_recycle_bin_at":null,\\"service_type":\\"EVS\","shareable":false,\\"size":40,\\"snapshot_id":null,\\"source_valid":null,\\"status":\\"in-use\","storage_cluster_id":\\"0\","sys_tags":\\"_sys_enterprise_project_id":\\"0\","tags":\{}\,"updated_at":\\"2019-05-10T03:52:41.053912\","user_id":\\"38d65be2ecd840d19046e239e841a734\","volume_image_metadata":\\"__account_code\","__backup_id":\\"\\","__data_origin":\\"\\","__description":\\"\\","__image_location":\\"192.149.48.66:443:pcsimssouthchina:79bee4ee-0025-4645-b004-23d2a66f6eec\","__image_size":\\"647303168\","__image_source_type":\\"uds\","__imagetype":\\"gold\","__isregistered":true\,"__lazyloading":true\,"__originalimagename":\\"79bee4ee-0025-4645-b004-23d2a66f6eec\","__os_bit":\\"64\","__os_type":\\"Linux\","__os_version":\\"CentOS 7.6 64bit\","__platform":\\"CentOS\","__productcode":\\"\\","__support_kvm":true\,"__support_xen":true\,"checksum":\\"99914b932bd37a50b983c5e7c90ae93b\","container_format":\\"bare\","disk_format":\\"vhdx\","image_id":\\"79bee4ee-0025-4645-b004-23d2a66f6eec\","image_name":\\"Public CentOS 7.6 64bit for Test\","min_disk":\\"40\","min_ram":\\"0\","size":\\"2\","virtual_env_type":\\"FusionCompute\\"",\\"volume_qos_specs":null,\\"volume_type":\\"SATA\\"",\\"interface":\\"{\\"interfaceAttachments":\[\{"port_state":\\"ACTIVE\","fixed_ips":\[\{"subnet_id":\\"63b881a8-daf0-45b1-90d8-e0c01a9e5275\","ip_address":\\"192.168.1.128\"\}],\\"port_id":\\"76a45e6e-8f02-4de2-9358-0110280ed000\","net_id":\\"0756f84d-03b3-44a0-95d8-c1ac70058f9d\","mac_addr":\\"fa:16:3e:ef:d5:fb\\"\}\}],\\"flavor":\\"{\\"flavor":\\"{\\"name":\\"s2.small.1\","links":\[\{"href":\\"https://compute.Region.dc1.domainname.com/v2/4229d7a45436489f8c3dc2b1d35d4987/flavors/s2.small.1\","rel":\\"self\","href":\\"https://compute.Region.dc1.domainname.com/4229d7a45436489f8c3dc2b1d35d4987/flavors/s2.small.1\","rel":\\"bookmark\\"\}],\\"ram":1024,\\"OS-FLV-DISABLED:disabled":false,\\"vcpus":1,\\"swap":\\"\\","os-flavor-access:is_public":true,\\"rxtx_factor":1.0,\\"OS-FLV-EXT-DATA:ephemeral":0,\\"disk":0,\\"id":\\"s2.small.1\\"",\\"ports":\[\{"port":\\"id":\\"76a45e6e-8f02-4de2-9358-0110280ed000\","name":\\"\\","admin_state_up":true,\\"status":\\"ACTIVE\","network_id":\\"0756f84d-03b3-44a0-95d8-c1ac70058f9d\","tenant_id":\\"4229d7a45436489f8c3dc2b1d35d4987\","project_id":\\"4229d7a45436489f8c3dc2b1d35d4987\","device_id":\\"94eba8b2-acc9-4d82-badc-127144cc5526\","mac_address":\\"fa:16:3e:ef:d5:fb\","device_owner":\\"compute:az1.dc1\","description":\\"\\","created_at":\\"2019-05-08T08:52:25\","updated_at":\\"2019-05-08T08:52:25\","port_security_enabled":true,\\"tags":\[],\\"security_groups":\[\\"a27d2564-b40c-41ec-9eb6-6eca3ff5aeee\"],\\"allowed_address_pairs":\[],\\"extra_dhcp_opts":\[],\\"fixed_ips":\[]}
```

```
[{"subnet_id": "63b881a8-daf0-45b1-90d8-e0c01a9e5275", "ip_address": "\\"192.168.1.128\\"], "dns_assignment": [{"ip_address": "\\"192.168.1.128\\", "hostname": "\\"ip-192-168-1-128\\", "fqdn": "\\"ip-192-168-1-128.southchina.compute.internal\\"}, {"qos_policy_id": "null", "dns_name": "\\"ip-192-168-1-128\\", "dns_domain": "\\"southchina.compute.internal\\", "instance_id": "\\"\\", "instance_type": "\\"\\", "ecs_flavor": "\\"\\", "binding:host_id": "\\"az1.dc1\\", "binding:vif_type": "\\"cascading\\", "binding:vnic_type": "\\"normal\\", "binding:vif_details": {"primary_interface": true}, "binding:profile": {}}], "floatingips": [ {"floatingips": "\\"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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowMetadataSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowMetadataRequest request = new ShowMetadataRequest();
        try {
            ShowMetadataResponse response = client.showMetadata(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowMetadataRequest()
        response = client.show_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"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMetadataRequest{}
    response, err := client.ShowMetadata(request)
```

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.3.8 更新备份

功能介绍

根据备份id更改备份

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/backups/{backup_id}

表 4-120 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份ID
project_id	是	String	项目ID

请求参数

表 4-121 请求 Body 参数

参数	是否必选	参数类型	描述
backup	否	BackupUpdate object	更新请求参数体

表 4-122 BackupUpdate

参数	是否必选	参数类型	描述
name	否	String	备份名称

响应参数

状态码： 200

表 4-123 响应 Body 参数

参数	参数类型	描述
backup	BackupResp object	备份副本

表 4-124 BackupResp

参数	参数类型	描述
checkpoint_id	String	还原点ID
created_at	String	创建时间，例如："2020-02-05T10:38:34.209782"
description	String	备份描述
expired_at	String	过期时间，例如："2020-02-05T10:38:34.209782"
extend_info	BackupExtendInfo object	扩展信息
id	String	备份ID
image_type	String	备份类型。取值为backup和replication。
name	String	备份名称
parent_id	String	父备份ID
project_id	String	项目ID

参数	参数类型	描述
protected_at	String	备份时间
resource_az	String	资源可用区
resource_id	String	资源ID
resource_name	String	资源名称
resource_size	Integer	资源大小，单位为GB
resource_type	String	资源类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
status	String	备份状态 枚举值: <ul style="list-style-type: none">• available• protecting• deleting• restoring• error• waiting_protect• waiting_delete• waiting_restore
updated_at	String	更新时间，例如:"2020-02-05T10:38:34.209782"
vault_id	String	存储库ID
replication_records	Array of ReplicationRecordGet objects	复制记录
enterprise_project_id	String	企业项目id,默认为‘0’。
provider_id	String	备份提供商ID，用于区分备份对象。当前取值包含： 0daac4c5-6707-4851-97ba-169e36266b66，该值代表备份对象为云服务器。 d1603440-187d-4516-af25-121250c7cc97，该值代表备份对象为云硬盘。 3f3c3220-245c-4805-b811-758870015881，该值代表备份对象为SFS Turbo。 a13639de-00be-4e94-af30-26912d75e4a2，该值代表备份对象为混合云VMware备份。

参数	参数类型	描述
children	Array of BackupResp objects	子副本列表

表 4-125 BackupExtendInfo

参数	参数类型	描述
auto_trigger	Boolean	是否是自动生成的备份副本
bootable	Boolean	是否系统盘备份
incremental	Boolean	是否是增备
snapshot_id	String	卷备份副本的快照id
support_lld	Boolean	是否支持lazyloading快速恢复
supported_restore_mode	String	备份支持恢复的方式，当前取值包含na,snapshot和backup。如果该字段取值为snapshot，代表备份此时已经支持创建整机镜像；如果该字段取值为backup，备份支持通过云服务器上硬盘的备份进行恢复；如果该字段取值为na，备份不支持直接恢复，只支持备份创建新资源等操作。 缺省值： na 枚举值： <ul style="list-style-type: none">• na• backup• snapshot
os_images_data	Array of ImageData objects	备份注册镜像ID列表
contain_system_disk	Boolean	整机备份是否包含系统盘
encrypted	Boolean	是否加密
system_disk	Boolean	是否是系统盘
is_multi_az	Boolean	备份类型是否为多AZ 缺省值： false

表 4-126 ImageData

参数	参数类型	描述
image_id	String	镜像ID

表 4-127 ReplicationRecordGet

参数	参数类型	描述
created_at	String	复制的开始时间
destination_backup_id	String	复制的目的备份ID
destination_checkpoint_id	String	复制的目的备份记录ID
destination_project_id	String	复制的目标项目ID
destination_region	String	复制的目标区域
destination_vault_id	String	目标存储库ID
extra_info	ReplicationRecordsExtraInfo object	复制附加信息
id	String	复制记录ID
source_backup_id	String	复制的源备份ID
source_checkpoint_id	String	复制的源备份记录ID
source_project_id	String	复制的源项目ID
source_region	String	复制的源区域
status	String	复制的状态 枚举值： <ul style="list-style-type: none">• replicating• success• fail• skip• waiting_replicate
vault_id	String	备份所在的存储库ID

表 4-128 ReplicationRecordsExtraInfo

参数	参数类型	描述
progress	Integer	复制进度
fail_code	String	失败错误码，成功时为空
fail_reason	String	错误原因
auto_trigger	Boolean	是否为自动调度复制
destination_vault_id	String	目标端的存储库id

请求示例

更改指定备份的备份名称。

```
PUT https://[endpoint]/v3/4229d7a45436489f8c3dc2b1d35d4987/backup/6df2b54c-dd62-4059-a07c-1b8f24f2725d

{
  "backup": {
    "name": "backup-name02"
  }
}
```

响应示例

状态码： 200

OK

```
{
  "backup": {
    "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",
    "checkpoint_id": "8b0851a8-adf3-4f4c-a914-dead08bf9664",
    "enterprise_project_id": 0,
    "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",
    "id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",
    "resource_az": "az1.dc1",
    "image_type": "backup",
    "resource_id": "94eba8b2-acc9-4d82-badc-127144cc5526",
    "resource_size": 40,
    "children": [
      {
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",
        "checkpoint_id": "8b0851a8-adf3-4f4c-a914-dead08bf9664",
        "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",
        "id": "5d822633-2bbf-4af8-a16e-5ab1c7705235",
        "image_type": "backup",
        "resource_id": "eccbcfdd-f843-4bbb-b2c0-a5ce861f9376",
        "resource_size": 40,
        "children": [],
        "parent_id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d",
        "extend_info": {
          "auto_trigger": true,
          "snapshot_id": "5230a977-1a94-4092-8edd-519303a44cda",
          "bootable": true,
          "encrypted": true
        },
        "project_id": "4229d7a45436489f8c3dc2b1d35d4987",
        "status": "available"
      }
    ]
  }
}
```

```
        "resource_name" : "ecs-1f0f-0002",
        "replication_records" : [ ],
        "name" : "autobk_a843_ecs-1f0f-0002",
        "created_at" : "2019-05-10T07:59:59.451+00:00",
        "resource_type" : "OS::Cinder::Volume"
    },
    "extend_info" : {
        "auto_trigger" : true,
        "supported_restore_mode" : "backup",
        "contain_system_disk" : true,
        "support_lld" : true
    },
    "project_id" : "4229d7a45436489f8c3dc2b1d35d4987",
    "status" : "available",
    "resource_name" : "ecs-1f0f-0002",
    "description" : "backup_description",
    "replication_records" : [ ],
    "name" : "backup-name02",
    "created_at" : "2019-05-10T07:59:12.085+00:00",
    "resource_type" : "OS::Nova::Server"
}
}
```

SDK 代码示例

SDK代码示例如下。

Java

更改指定备份的备份名称。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class UpdateBackupSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateBackupRequest request = new UpdateBackupRequest();
        BackupUpdateReq body = new BackupUpdateReq();
        BackupUpdate backupbody = new BackupUpdate();
        backupbody.withName("backup-name02");
        body.withBackup(backupbody);
    }
}
```

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

Python

更改指定备份的备份名称。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateBackupRequest()
        backupbody = BackupUpdate(
            name="backup-name02"
        )
        request.body = BackupUpdateReq(
            backup=backupbody
        )
        response = client.update_backup(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更改指定备份的备份名称。

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateBackupRequest{}
    nameBackup:= "backup-name02"
    backupbody := &model.BackupUpdate{
        Name: &nameBackup,
    }
    request.Body = &model.BackupUpdateReq{
        Backup: backupbody,
    }
    response, err := client.UpdateBackup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.4 备份共享

4.4.1 添加备份成员

功能介绍

添加备份可共享的成员，只有云服务器和云硬盘备份可以添加备份共享成员，支持不同项目之间共享和不同用户之间共享。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/backups/{backup_id}/members

表 4-129 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份副本id
project_id	是	String	项目ID

请求参数

表 4-130 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-131 请求 Body 参数

参数	是否必选	参数类型	描述
members	是	Array of strings	列表，待添加备份共享成员的project_id。 数组长度：1 - 10

响应参数

状态码： 200

表 4-132 响应 Body 参数

参数	参数类型	描述
members	Array of Member objects	添加备份共享成员响应信息
count	Integer	备份共享成员数量

表 4-133 Member

参数	参数类型	描述
status	String	共享状态 枚举值： <ul style="list-style-type: none">• pending• accepted• rejected
created_at	String	共享时间，例如："2020-02-05T10:38:34.209782"
updated_at	String	更新时间，例如："2020-02-05T10:38:34.209782"
backup_id	String	备份副本id
image_id	String	接受的共享备份副本注册的镜像id
dest_project_id	String	接受备份共享的项目id
vault_id	String	目标端接受共享备份的存储库id
id	String	共享记录id

请求示例

添加备份可共享的成员。

```
POST https://{{endpoint}}/v3/0605767b5780d5762fc5c0118072a564/backups/  
0b07081e-3ec7-4e77-8571-54e2947da422/members
```

```
{  
  "members" : [ "075e6035d300d48c2fd0c00b78b71ebf" ]  
}
```

响应示例

状态码： 200

OK

```
{  
  "members" : [ {
```

```
        "status" : "pending",
        "backup_id" : "0b07081e-3ec7-4e77-8571-54e2947da422",
        "dest_project_id" : "075e6035d300d48c2fd0c00b78b71ebf",
        "created_at" : "2020-02-05T10:38:34.210+00:00",
        "id" : "3c5a3015-c3a0-4dc6-a1e2-917b90f62319"
    } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

添加备份可共享的成员。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class AddMemberSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        AddMemberRequest request = new AddMemberRequest();
        AddMembersReq body = new AddMembersReq();
        List<String> listbodyMembers = new ArrayList<>();
        listbodyMembers.add("075e6035d300d48c2fd0c00b78b71ebf");
        body.withMembers(listbodyMembers);
        request.withBody(body);
        try {
            AddMemberResponse response = client.addMember(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

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

Python

添加备份可共享的成员。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddMemberRequest()
        listMembersbody = [
            "075e6035d300d48c2fd0c00b78b71ebf"
        ]
        request.body = AddMembersReq(
            members=listMembersbody
        )
        response = client.add_member(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

添加备份可共享的成员。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>").
        WithCredential(auth).
        Build())

request := &model.AddMemberRequest{}
var listMembersbody = []string{
    "075e6035d300d48c2fd0c00b78b71ebf",
}
request.Body = &model.AddMembersReq{
    Members: listMembersbody,
}
response, err := client.AddMember(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.4.2 更新备份成员状态

功能介绍

更新备份共享成员的状态，需要接收方执行此API。

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/backups/{backup_id}/members/{member_id}

表 4-134 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份副本id
member_id	是	String	成员id，成员id与项目id为同一个。
project_id	是	String	项目id

请求参数

表 4-135 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-136 请求 Body 参数

参数	是否必选	参数类型	描述
status	是	String	备份共享状态 枚举值： • accepted • pending • rejected
vault_id	否	String	共享的备份将存入的存储库，仅支持uuid 更新member状态的时候，如果是接受，必须传入 vault_id，如果是拒绝，则无需

响应参数

状态码： 200

表 4-137 响应 Body 参数

参数	参数类型	描述
member	Member object	查询备份共享成员详情响应信息

表 4-138 Member

参数	参数类型	描述
status	String	共享状态 枚举值： <ul style="list-style-type: none">• pending• accepted• rejected
created_at	String	共享时间，例如："2020-02-05T10:38:34.209782"
updated_at	String	更新时间，例如："2020-02-05T10:38:34.209782"
backup_id	String	备份副本id
image_id	String	接受的共享备份副本注册的镜像id
dest_project_id	String	接受备份共享的项目id
vault_id	String	目标端接受共享备份的存储库id
id	String	共享记录id

请求示例

```
PUT https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/backups/a5200268-74a5-4806-acc6-95793ab0228b/members/f841e01fd2b14e7fa41b6ae7aa6b0594

{
    "status" : "accepted",
    "vault_id" : "4b27c05b-8ad7-48c6-a886-526666c035f0"
}
```

响应示例

状态码： 200

OK

```
{
    "member" : {
        "status" : "accepted",
        "backup_id" : "17c9acd8-3af3-4401-bab9-ff1cfac15561",
        "vault_id" : "4b27c05b-8ad7-48c6-a886-526666c035f0",
        "dest_project_id" : "0761021b8900d2622f38c0115db0b331",
        "created_at" : "2020-02-24T09:36:00.479+00:00",
        "id" : "824a90b3-c562-448b-ab04-60ea4a97cf60"
    }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class UpdateMemberStatusSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateMemberStatusRequest request = new UpdateMemberStatusRequest();
        UpdateMember body = new UpdateMember();
        body.withVaultId("4b27c05b-8ad7-48c6-a886-526666c035f0");
        body.withStatus(UpdateMember.StatusEnum.fromValue("accepted"));
        request.withBody(body);
        try {
            UpdateMemberStatusResponse response = client.updateMemberStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = UpdateMemberStatusRequest()
    request.body = UpdateMember(
        vault_id="4b27c05b-8ad7-48c6-a886-526666c035f0",
        status="accepted"
    )
    response = client.update_member_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"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateMemberStatusRequest{}
    vaultIdUpdateMember:= "4b27c05b-8ad7-48c6-a886-526666c035f0"
    request.Body = &model.UpdateMember{
        VaultId: &vaultIdUpdateMember,
        Status: model.GetUpdateMemberStatusEnum().ACCEPTED,
    }
    response, err := client.UpdateMemberStatus(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.4.3 获取备份成员详情

功能介绍

获取备份成员的详情

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/backups/{backup_id}/members/{member_id}

表 4-139 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份副本id
member_id	是	String	成员id，为接收方的project_id
project_id	是	String	项目id

请求参数

表 4-140 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-141 响应 Body 参数

参数	参数类型	描述
member	Member object	查询备份共享成员详情响应信息

表 4-142 Member

参数	参数类型	描述
status	String	共享状态 枚举值： <ul style="list-style-type: none">• pending• accepted• rejected
created_at	String	共享时间，例如："2020-02-05T10:38:34.209782"
updated_at	String	更新时间，例如："2020-02-05T10:38:34.209782"
backup_id	String	备份副本id
image_id	String	接受的共享备份副本注册的镜像id
dest_project_id	String	接受备份共享的项目id
vault_id	String	目标端接受共享备份的存储库id
id	String	共享记录id

请求示例

```
GET https://{{endpoint}}/v3/{{project_id}}/backups/{{backup_id}}/members/{{member_id}}
```

响应示例

状态码： 200

OK

```
{  
    "member": {  
        "status": "accepted",  
        "backup_id": "17c9acd8-3af3-4401-bab9-ff1cfac15561",  
        "vault_id": "4b27c05b-8ad7-48c6-a886-526666c035f0",  
        "dest_project_id": "0761021b8900d2622f38c0115db0b331",  
        "created_at": "2020-02-24T09:36:00.479+00:00",  
        "id": "824a90b3-c562-448b-ab04-60ea4a97cf60"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowMemberDetailSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowMemberDetailRequest request = new ShowMemberDetailRequest();  
        try {  
            ShowMemberDetailResponse response = client.showMemberDetail(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

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

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
    cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ShowMemberDetailRequest{}  
response, err := client.ShowMemberDetail(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.4.4 获取备份成员列表

功能介绍

获取备份共享成员的列表信息

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/backups/{backup_id}/members

表 4-143 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份id
project_id	是	String	项目id

表 4-144 Query 参数

参数	是否必选	参数类型	描述
dest_project_id	否	String	接受备份共享的项目id
image_id	否	String	接受的共享备份副本注册的镜像id
limit	否	Integer	每页显示的条目数量，正整数
marker	否	String	上一次查询最后一条的id，仅支持uuid
offset	否	Integer	偏移值，正整数
sort	否	String	sort的内容为一组由逗号分隔的属性及可选排序方向组成，形如<key1>[:<direction>],<key2>[:<direction>],其中direction的取值为asc(升序)或desc(降序)，如没有传入direction参数，默认为降序，sort内容的长度限制为255个字符。
status	否	String	备份共享状态
vault_id	否	String	目标端接受共享备份的存储库id，仅支持uuid

请求参数

表 4-145 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取(响应消息头中X-Subject-Token的值)。

响应参数

状态码： 200

表 4-146 响应 Body 参数

参数	参数类型	描述
members	Array of Member objects	添加备份共享成员响应信息
count	Integer	备份共享成员数量

表 4-147 Member

参数	参数类型	描述
status	String	共享状态 枚举值： <ul style="list-style-type: none">• pending• accepted• rejected
created_at	String	共享时间，例如："2020-02-05T10:38:34.209782"
updated_at	String	更新时间，例如："2020-02-05T10:38:34.209782"
backup_id	String	备份副本id
image_id	String	接受的共享备份副本注册的镜像id
dest_project_id	String	接受备份共享的项目id
vault_id	String	目标端接受共享备份的存储库id
id	String	共享记录id

请求示例

```
GET https://{endpoint}/v3/{project_id}/backups/{backup_id}/members
```

响应示例

状态码： 200

OK

```
{  
  "members": [ {  
    "status": "pending",  
    "backup_id": "0b07081e-3ec7-4e77-8571-54e2947da422",  
    "dest_project_id": "075e6035d300d48c2fd0c00b78b71ebf",  
    "created_at": "2020-02-05T10:38:34.210+00:00",  
    "id": "3c5a3015-c3a0-4dc6-a1e2-917b90f62319"  
  } ],  
  "count": 50  
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowMembersDetailSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowMembersDetailRequest request = new ShowMembersDetailRequest();
        request.withDestProjectId("<dest_project_id>");
        request.withImageId("<image_id>");
        request.withStatus("<status>");
        request.withVaultId("<vault_id>");
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withOffset(<offset>);
        request.withSort("<sort>");
        try {
            ShowMembersDetailResponse response = client.showMembersDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowMembersDetailRequest()
        request.dest_project_id = "<dest_project_id>"
        request.image_id = "<image_id>"
        request.status = "<status>"
        request.vault_id = "<vault_id>"
        request.limit = <limit>
        request.marker = "<marker>"
        request.offset = <offset>
        request.sort = "<sort>"
        response = client.show_members_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region" "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ShowMembersDetailRequest{}
destProjectIdRequest:= "<dest_project_id>"
request.DestProjectId = &destProjectIdRequest
imageIdRequest:= "<image_id>"
request.ImageId = &imageIdRequest
statusRequest:= "<status>"
request.Status = &statusRequest
vaultIdRequest:= "<vault_id>"
request.VaultId = &vaultIdRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
markerRequest:= "<marker>"
request.Marker = &markerRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
sortRequest:= "<sort>"
request.Sort = &sortRequest
response, err := client.ShowMembersDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.4.5 删除指定备份成员

功能介绍

删除指定的备份共享成员

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/backups/{backup_id}/members/{member_id}

表 4-148 路径参数

参数	是否必选	参数类型	描述
backup_id	是	String	备份副本id
member_id	是	String	成员id
project_id	是	String	项目id

请求参数

表 4-149 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

无

请求示例

```
DELETE https://{{endpoint}}/v3/{{project_id}}/backups/{{backup_id}}/members/{{member_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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DeleteMemberSolution {

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

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

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteMemberRequest request = new DeleteMemberRequest();
try {
    DeleteMemberResponse response = client.deleteMember(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \
        client = CbrClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \
            .build()

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

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

更多

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

状态码

状态码	描述
204	No Content

错误码

请参见[错误码](#)。

4.5 存储库

4.5.1 创建存储库

功能介绍

创建存储库

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults

表 4-150 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-151 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-152 请求 Body 参数

参数	是否必选	参数类型	描述
vault	是	VaultCreate object	存储库创建参数

表 4-153 VaultCreate

参数	是否必选	参数类型	描述
backup_policy_id	否	String	备份策略ID，不设置时为null，不自动备份。
billing	是	BillingCreate object	创建参数信息

参数	是否必选	参数类型	描述
description	否	String	描述 最小长度: 0 最大长度: 64
name	是	String	存储库名称 最小长度: 1 最大长度: 64
resources	是	Array of ResourceCreate objects	绑定的备份资源，未在创建时绑定资源填[] 数组长度: 0 - 255
tags	否	Array of Tag objects	标签列表 tags不允许为空列表。tags中最多包含10个key。tags中key不允许重复。
enterprise_project_id	否	String	企业项目ID， 默认为 ‘0’ 。
auto_bind	否	Boolean	是否支持自动挂载。
bind_rules	否	VaultBindRules object	自动挂载的规则
auto_expand	否	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
threshold	否	Integer	存储库容量阈值，已用容量占总容量达到此百分比，将根据smn_notify 参数设置选择是否发送相关通知。默认值为: 80 最大值: 100 最小值: 1 最小值: 1 最大值: 100 缺省值: 80
smn_notify	否	Boolean	存储库smn消息通知开关。默认值为 true。 缺省值: true
backup_name_prefix	否	String	备份名称前缀，设置后该存储库自动备份产生的备份副本都将携带该备份名称前缀 最小长度: 0 最大长度: 32
demand_billing	否	Boolean	存储库使用是否允许超出容量，只有创建包周期存储库时才允许该值为 true 缺省值: false

参数	是否必选	参数类型	描述
sys_lock_source_service	否	String	用于标识SMB服务， SMB可以设置sys_lock_source_service标签 缺省值： "" 枚举值： <ul style="list-style-type: none">• SMB• "

表 4-154 BillingCreate

参数	是否必选	参数类型	描述
cloud_type	否	String	公有云:public ;混合云:hybrid
consistent_level	是	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	是	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
protect_type	是	String	保护类型: 备份 (backup)、复制(replication)。
size	是	Integer	容量, 单位GB 最小值: 10 最大值: 10485760
charging_mode	否	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为 post_paid 缺省值: post_paid
period_type	否	String	创建类型, charging_mode为 pre_paid必填, 按年(year)或者按月(month) 枚举值： <ul style="list-style-type: none">• year• month
period_num	否	Integer	创建类型的数量, charging_mode为pre_paid必填

参数	是否必选	参数类型	描述
is_auto_renew	否	Boolean	到期后是否自动续期， 默认不续期 缺省值： false
is_auto_pay	否	Boolean	是否自动付费， 默认为不自动付费 缺省值： false
console_url	否	String	跳转URL 最小长度： 1 最大长度： 255
is_multi_az	否	Boolean	存储库多az属性， 默认为false 缺省值： false

表 4-155 ResourceCreate

参数	是否必选	参数类型	描述
extra_info	否	ResourceExtraInfo object	资源附加信息
id	是	String	待备份资源id
type	是	String	待备份资源的类型： OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
name	否	String	名称 最小长度： 0 最大长度： 255

表 4-156 ResourceExtraInfo

参数	是否必选	参数类型	描述
exclude_volumes	否	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有效，排除不需要备份的磁盘。当虚拟机新绑定磁盘时，也能继续排除之前设置不用备份的卷。

表 4-157 Tag

参数	是否必选	参数类型	描述
key	是	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	是	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-158 VaultBindRules

参数	是否必选	参数类型	描述
tags	否	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-159 BindRulesTags

参数	是否必选	参数类型	描述
key	是	String	key不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	是	String	value不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

响应参数

状态码: 200

表 4-160 响应 Body 参数

参数	参数类型	描述
vault	VaultCreateResource object	存储库查询返回对象

表 4-161 VaultCreateResource

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id

参数	参数类型	描述
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’ 。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80
errText	String	包周期创建错误信息
retCode	String	包周期订购结果
orders	Array of CbcOrderRes ult objects	包周期创建订单信息
backup_name_prefix	String	备份名称前缀 最小长度: 0 最大长度: 32
demand_billing	Boolean	是否允许使用超出存储库容量 缺省值: false
cbc_delete_count	Integer	存储库删除次数 缺省值: 0
frozen	Boolean	存储库是否冻结 缺省值: false

表 4-162 Billing

参数	参数类型	描述
allocated	Integer	已分配容量, 单位GB
charging_mod e	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal
status	String	存储库状态 枚举值: <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-163 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值: <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小
backup_count	Integer	副本数量

表 4-164 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效, 排除不需要备份的磁盘。当虚拟机新绑定磁 盘时, 也能继续排除之前设置不用备份的卷。

表 4-165 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-166 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-167 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。

参数	参数类型	描述
value	String	value不能包含非打印字符ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 value只能由中文, 字母, 数字, “-”, “_”, “.” 组成。

表 4-168 CbcOrderResult

参数	参数类型	描述
cloudServiceId	String	云服务ID
orderId	String	订单ID
subscribeResult	Integer	订购结果, 1: 成功; 0: 失败
resourceId	String	包周期资源预生成资源id。

状态码: 400

表 4-169 响应 Body 参数

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

请求示例

- 创建一个云服务器存储库，存储库容量为100G，按需计费，且同时绑定资源和添加标签。

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults
{
    "vault": {
        "backup_policy_id": "6dd81d7d-a4cb-443e-b8ed-1af0bd3a261b",
        "billing": {
            "cloud_type": "public",
            "consistent_level": "crash_consistent",
            "object_type": "server",
            "protect_type": "backup",
            "size": 100,
            "charging_mode": "post_paid",
            "is_auto_renew": false,
            "is_auto_pay": false,
            "console_url": "https://console.demo.com/cbr/?agencyId=97fc896b7914cb98f553a087232e243&region=testregion/cbr/manager/csbs/vaultList"
        },
        "description": "vault_description",
        "name": "vault_name",
    }
}
```

```
"resources" : [ {
    "extra_info" : {
        "exclude_volumes" : [ "43a320a5-3efd-4568-b1aa-8dd9183cc64b" ]
    },
    "id" : "23a320a5-3efd-4568-b1aa-8dd9183cc64c",
    "type" : "OS::Nova::Server"
} ],
"tags" : [ {
    "key" : "key01",
    "value" : "value01"
} ],
"enterprise_project_id" : "0"
}
```

- 创建一个云硬盘备份存储库，存储库容量为40G,按需计费。

POST <https://{{endpoint}}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults>

```
{
    "vault" : {
        "billing" : {
            "consistent_level" : "crash_consistent",
            "object_type" : "disk",
            "protect_type" : "backup",
            "size" : 40,
            "charging_mode" : "post_paid",
            "is_auto_renew" : false,
            "is_auto_pay" : false
        },
        "name" : "test",
        "resources" : [ ]
    }
}
```

响应示例

状态码： 200

OK

```
{
    "vault" : {
        "provider_id" : "0daac4c5-6707-4851-97ba-169e36266b66",
        "description" : "vault_description",
        "tags" : [ {
            "value" : "value01",
            "key" : "key01"
        } ],
        "enterprise_project_id" : 0,
        "auto_bind" : false,
        "id" : "ad7627ae-5b0b-492e-b6bd-cd809b745197",
        "user_id" : "38d65be2ecd840d19046e239e841a734",
        "name" : "vault_name",
        "billing" : {
            "status" : "available",
            "used" : 0,
            "protect_type" : "backup",
            "object_type" : "server",
            "allocated" : 40,
            "spec_code" : "vault.backup.server.normal",
            "size" : 100,
            "cloud_type" : "public",
            "consistent_level" : "crash_consistent",
            "charging_mode" : "post_paid"
        },
        "created_at" : "2019-05-23T12:51:10.071+00:00",
        "project_id" : "fc347bc64ccd4589ae52e4f44b7433c7",
        "resources" : [ {
```

```
        "name" : "ecs-b977-0002",
        "backup_size" : 0,
        "protect_status" : "available",
        "backup_count" : 0,
        "extra_info" : {
            "exclude_volumes" : [ "1855eb9a-2b5e-4938-a9f0-aea08b6f9243", "5a51e8b3-2f65-4045-896f-f8ffae14b064" ]
        },
        "type" : "OS::Nova::Server",
        "id" : "23a320a5-3efd-4568-b1aa-8dd9183cc64c",
        "size" : 40
    }
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建一个云服务器存储库，存储库容量为100G，按需计费，且同时绑定资源和添加标签。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class CreateVaultSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateVaultRequest request = new CreateVaultRequest();
        VaultCreateReq body = new VaultCreateReq();
        List<Tag> listVaultTags = new ArrayList<>();
        listVaultTags.add(
            new Tag()
                .withKey("key01")
                .withValue("value01")
        );
        List<String> listExtraInfoExcludeVolumes = new ArrayList<>();
        listExtraInfoExcludeVolumes.add("43a320a5-3efd-4568-b1aa-8dd9183cc64b");
    }
}
```

```
ResourceExtraInfo extraInfoResources = new ResourceExtraInfo();
extraInfoResources.withExcludeVolumes(listExtraInfoExcludeVolumes);
List<ResourceCreate> listVaultResources = new ArrayList<>();
listVaultResources.add(
    new ResourceCreate()
        .withExtraInfo(extraInfoResources)
        .withId("23a320a5-3efd-4568-b1aa-8dd9183cc64c")
        .withType("OS::Nova::Server")
);
BillingCreate billingVault = new BillingCreate();
billingVault.withCloudType(BillingCreate.CloudTypeEnum.fromValue("public"))
    .withConsistentLevel(BillingCreate.ConsistentLevelEnum.fromValue("crash_consistent"))
    .withObjectType(BillingCreate.ObjectTypeEnum.fromValue("server"))
    .withProtectType(BillingCreate.ProtectTypeEnum.fromValue("backup"))
    .withSize(100)
    .withChargingMode(BillingCreate.ChargingModeEnum.fromValue("post_paid"))
    .withIsAutoRenew(false)
    .withIsAutoPay(false)
    .withConsoleUrl("https://console.demo.com/cbr/?"
agencyId=97fc9896b7914cb98f553a087232e243&region=testregion/cbr/manager/csbs/vaultList");
VaultCreate vaultbody = new VaultCreate();
vaultbody.withBackupPolicyId("6dd81d7d-a4cb-443e-b8ed-1af0bd3a261b")
    .withBilling(billingVault)
    .withDescription("vault_description")
    .withName("vault_name")
    .withResources(listVaultResources)
    .withTags(listVaultTags)
    .withEnterpriseProjectId("0");
body.withVault(vaultbody);
request.withBody(body);
try {
    CreateVaultResponse response = client.createVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 创建一个云硬盘备份存储库，存储库容量为40G,按需计费。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class CreateVaultSolution {

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

    CbrClient client = CbrClient.newBuilder()
        .withCredential(auth)
        .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateVaultRequest request = new CreateVaultRequest();
    VaultCreateReq body = new VaultCreateReq();
    BillingCreate billingVault = new BillingCreate();

    billingVault.withConsistentLevel(BillingCreate.ConsistentLevelEnum.fromValue("crash_consistent"))
        .withObjectType(BillingCreate.ObjectTypeEnum.fromValue("disk"))
        .withProtectType(BillingCreate.ProtectTypeEnum.fromValue("backup"))
        .withSize(40)
        .withChargingMode(BillingCreate.ChargingModeEnum.fromValue("post_paid"))
        .withIsAutoRenew(false)
        .withIsAutoPay(false);
    VaultCreate vaultbody = new VaultCreate();
    vaultbody.withBilling(billingVault)
        .withName("test");
    body.withVault(vaultbody);
    request.withBody(body);
    try {
        CreateVaultResponse response = client.createVault(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- 创建一个云服务器存储库，存储库容量为100G，按需计费，且同时绑定资源和添加标签。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

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

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

try:
    request = CreateVaultRequest()
    listTagsVault = [
        Tag(
            key="key01",
            value="value01"
        )
    ]
    listExcludeVolumesExtraInfo = [
        "43a320a5-3efd-4568-b1aa-8dd9183cc64b"
    ]
    extraInfoResources = ResourceExtraInfo(
        exclude_volumes=listExcludeVolumesExtraInfo
    )
    listResourcesVault = [
        ResourceCreate(
            extra_info=extraInfoResources,
            id="23a320a5-3efd-4568-b1aa-8dd9183cc64c",
            type="OS::Nova::Server"
        )
    ]
    billingVault = BillingCreate(
        cloud_type="public",
        consistent_level="crash_consistent",
        object_type="server",
        protect_type="backup",
        size=100,
        charging_mode="post_paid",
        is_auto_renew=False,
        is_auto_pay=False,
        console_url="https://console.demo.com/cbr/?agencyId=97fcdb896b7914cb98f553a087232e243&region=testregion/cbr/manager/csbs/vaultList"
    )
    vaultbody = VaultCreate(
        backup_policy_id="6dd81d7d-a4cb-443e-b8ed-1af0bd3a261b",
        billing=billingVault,
        description="vault_description",
        name="vault_name",
        resources=listResourcesVault,
        tags=listTagsVault,
        enterprise_project_id="0"
    )
    request.body = VaultCreateReq(
        vault=vaultbody
    )
    response = client.create_vault(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建一个云硬盘备份存储库，存储库容量为40G,按需计费。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = CreateVaultRequest()
    billingVault = BillingCreate(
        consistent_level="crash_consistent",
        object_type="disk",
        protect_type="backup",
        size=40,
        charging_mode="post_paid",
        is_auto_renew=False,
        is_auto_pay=False
    )
    vaultbody = VaultCreate(
        billing=billingVault,
        name="test"
    )
    request.body = VaultCreateReq(
        vault=vaultbody
    )
    response = client.create_vault(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- 创建一个云服务器存储库，存储库容量为100G，按需计费，且同时绑定资源和添加标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region" "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>").
        WithCredential(auth).
        Build())

request := &model.CreateVaultRequest{}
var listTagsVault = []model.Tag{
    {
        Key: "key01",
        Value: "value01",
    },
}
var listExcludeVolumesExtraInfo = []string{
    "43a320a5-3efd-4568-b1aa-8dd9183cc64b",
}
extraInfoResources := &model.ResourceExtraInfo{
    ExcludeVolumes: &listExcludeVolumesExtraInfo,
}
var listResourcesVault = []model.ResourceCreate{
    {
        ExtraInfo: extraInfoResources,
        Id: "23a320a5-3efd-4568-b1aa-8dd9183cc64c",
        Type: "OS::Nova::Server",
    },
}
cloudTypeBilling:= model.GetBillingCreateCloudTypeEnum().PUBLIC
chargingModeBilling:= model.GetBillingCreateChargingModeEnum().POST_PAID
isAutoRenewBilling:= false
isAutoPayBilling:= false
consoleUrlBilling:= "https://console.demo.com/cbr/?"
agencyId=97fc896b7914cb98f553a087232e243&region=testregion/cbr/manager/csbs/vaultList"
billingVault := &model.BillingCreate{
    CloudType: &cloudTypeBilling,
    ConsistentLevel: model.GetBillingCreateConsistentLevelEnum().CRASH_CONSISTENT,
    ObjectType: model.GetBillingCreateObjectTypeEnum().SERVER,
    ProtectType: model.GetBillingCreateProtectTypeEnum().BACKUP,
    Size: int32(100),
    ChargingMode: &chargingModeBilling,
    IsAutoRenew: &isAutoRenewBilling,
    IsAutoPay: &isAutoPayBilling,
    ConsoleUrl: &consoleUrlBilling,
}
backupPolicyIdVault:= "6dd81d7d-a4cb-443e-b8ed-1af0bd3a261b"
descriptionVault:= "vault_description"
enterpriseProjectIdVault:= "0"
vaultbody := &model.VaultCreate{
    BackupPolicyId: &backupPolicyIdVault,
    Billing: billingVault,
    Description: &descriptionVault,
    Name: "vault_name",
    Resources: listResourcesVault,
    Tags: &listTagsVault,
    EnterpriseProjectId: &enterpriseProjectIdVault,
}
request.Body = &model.VaultCreateReq{
    Vault: vaultbody,
}
response, err := client.CreateVault(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 创建一个云硬盘备份存储库，存储库容量为40G,按需计费。

```
package main
```

```
import (
```

```
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.CreateVaultRequest{}
    chargingModeBilling:= model.GetBillingCreateChargingModeEnum().POST_PAID
    isAutoRenewBilling:= false
    isAutoPayBilling:= false
    billingVault := &model.BillingCreate{
        ConsistentLevel: model.GetBillingCreateConsistentLevelEnum().CRASH_CONSISTENT,
        ObjectType: model.GetBillingCreateObjectTypeEnum().DISK,
        ProtectType: model.GetBillingCreateProtectTypeEnum().BACKUP,
        Size: int32(40),
        ChargingMode: &chargingModeBilling,
        IsAutoRenew: &isAutoRenewBilling,
        IsAutoPay: &isAutoPayBilling,
    }
    vaultbody := &model.VaultCreate{
        Billing: billingVault,
        Name: "test",
    }
    request.Body = &model.VaultCreateReq{
        Vault: vaultbody,
    }
    response, err := client.CreateVault(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request

错误码

请参见[错误码](#)。

4.5.2 查询指定存储库

功能介绍

根据ID查询指定存储库

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vaults/{vault_id}

表 4-170 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-171 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-172 响应 Body 参数

参数	参数类型	描述
vault	Vault object	存储库查询返回对象

表 4-173 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true

参数	参数类型	描述
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值：1 最大值：100 缺省值：80
sys_lock_source_service	String	用于标识SMB服务 最小长度：0 最大长度：32 枚举值： <ul style="list-style-type: none">• SMB• "

表 4-174 Billing

参数	参数类型	描述
allocated	Integer	已分配容量，单位GB
charging_mode	String	创建模式，按需：post_paid，包周期：pre_paid，默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格，崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型：云服务器 (server)，云硬盘 (disk)，文件系统 (turbo)，云桌面 (workspace)，VMware (vmware)，关系型数据库 (rds)，文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型：备份 (backup)、复制 (replication)。
size	Integer	容量，单位GB 最小值：1 最大值：10485760
spec_code	String	规格编码。云服务备份存储库：vault.backup.server.normal；云硬盘备份存储库：vault.backup.volume.normal；文件备份存储库：vault.backup.turbo.normal

参数	参数类型	描述
status	String	存储库状态 枚举值： <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-175 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小

参数	参数类型	描述
backup_count	Integer	副本数量

表 4-176 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-177 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可 选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-178 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-179 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

状态码: 404

表 4-180 响应 Body 参数

参数	参数类型	描述
vault	Vault object	存储库查询返回对象

表 4-181 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64

参数	参数类型	描述
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_pro ject_id	String	企业项目id， 默认为‘0’。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRul es object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储 库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分 比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80
sys_lock_sourc e_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值: <ul style="list-style-type: none">• SMB• "

表 4-182 Billing

参数	参数类型	描述
allocated	Integer	已分配容量，单位GB
charging_mod e	String	创建模式，按需: post_paid，包周期: pre_paid， 默认为post_paid

参数	参数类型	描述
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal
status	String	存储库状态 枚举值: <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-183 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息

参数	参数类型	描述
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值: <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小
backup_count	Integer	副本数量

表 4-184 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效, 排除不需要备份的磁盘。当虚拟机新绑定磁 盘时, 也能继续排除之前设置不用备份的卷。

表 4-185 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 key只能由中文, 字母, 数字, “-”, “_”组成。

参数	参数类型	描述
value	String	值。 添加标签时value值必选，删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-186 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-187 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

查询指定存储库。

```
GET https://{{endpoint}}/v3/{{project_id}}/vaults/{{vault_id}}
```

响应示例

状态码: 200

OK

```
{  
    "vault": {  
        "id": "2b076f26-391f-40fa-bf71-a798940faccf",  
        "name": "sdk-vault1595581958",  
        "resources": [],  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "created_at": "2020-07-24T09:12:49.492+00:00",  
        "project_id": "0605767b5780d5762fc5c0118072a564",  
        "enterprise_project_id": 0,  
        "auto_bind": true,  
        "bind_rules": {},  
        "auto_expand": false,  
        "user_id": "aa2999fa5ae640f28926f8fd79188934",  
        "billing": {  
            "allocated": 0,  
            "cloud_type": "public",  
            "consistent_level": "crash_consistent",  
            "charging_mode": "post_paid",  
            "protect_type": "backup",  
            "object_type": "server",  
            "spec_code": "vault.backup.server.normal",  
            "used": 0,  
            "status": "available",  
            "size": 40  
        },  
        "tags": [ {  
            "key": "fds"  
        } ]  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowVaultSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)
```

```
.withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ShowVaultRequest request = new ShowVaultRequest();
try {
    ShowVaultResponse response = client.showVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>").
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
200	OK
404	Not Found

错误码

请参见[错误码](#)。

4.5.3 查询存储库列表

功能介绍

查询存储库列表

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vaults

表 4-188 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-189 Query 参数

参数	是否必选	参数类型	描述
cloud_type	否	String	公有云:public ;混合云:hybrid
enterprise_project_id	否	String	企业项目id或all_granted_eps, all_granted_eps表示查询用户有权限的所有企业项目id
id	否	String	存储库ID
limit	否	Integer	每页显示条目数, 正整数 最小值: 1 最大值: 1000
name	否	String	存储库名称
object_type	否	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
offset	否	Integer	偏移值,正整数
policy_id	否	String	策略ID
protect_type	否	String	保护类型: 备份 (backup)、复制(replication)。
resource_ids	否	String	资源id, 支持多资源, 以英文逗号分割
status	否	String	状态

请求参数

表 4-190 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-191 响应 Body 参数

参数	参数类型	描述
vaults	Array of Vault objects	存储库实例列表
count	Integer	存储库个数
limit	Integer	每页显示的条目数量 最小值: 1 最大值: 1000 缺省值: 1000
offset	Integer	偏移量，表示从此偏移量开始查询 最小值: 0 缺省值: 0

表 4-192 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64

参数	参数类型	描述
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_pro ject_id	String	企业项目id， 默认为‘0’。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRul es object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储 库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分 比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80
sys_lock_sourc e_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值: <ul style="list-style-type: none">• SMB• "

表 4-193 Billing

参数	参数类型	描述
allocated	Integer	已分配容量，单位GB
charging_mod e	String	创建模式，按需: post_paid，包周期: pre_paid， 默认为post_paid

参数	参数类型	描述
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal
status	String	存储库状态 枚举值: <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-194 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息

参数	参数类型	描述
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值: <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小
backup_count	Integer	副本数量

表 4-195 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-196 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 key只能由中文, 字母, 数字, “-”, “_”组成。

参数	参数类型	描述
value	String	值。 添加标签时value值必选，删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-197 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-198 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

查询存储库列表。

```
GET https://{{endpoint}}/v3/{{project_id}}/vaults
```

响应示例

状态码: 200

OK

```
{  
    "vaults" : [ {  
        "id" : "a335f9e1-1628-4c64-a7be-38656e5ec19c",  
        "name" : "vault-8538",  
        "resources" : [ ],  
        "provider_id" : "0daac4c5-6707-4851-97ba-169e36266b66",  
        "created_at" : "2020-09-04T06:57:37.344+00:00",  
        "project_id" : "0605767b5780d5762fc5c0118072a564",  
        "enterprise_project_id" : 0,  
        "auto_bind" : false,  
        "bind_rules" : { },  
        "auto_expand" : false,  
        "user_id" : "aa2999fa5ae640f28926f8fd79188934",  
        "billing" : {  
            "allocated" : 0,  
            "cloud_type" : "public",  
            "consistent_level" : "crash_consistent",  
            "charging_mode" : "post_paid",  
            "protect_type" : "backup",  
            "object_type" : "server",  
            "spec_code" : "vault.backup.server.normal",  
            "used" : 0,  
            "status" : "available",  
            "size" : 100  
        },  
        "tags" : [ ]  
    } ],  
    "count" : 50  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListVaultSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()
```

```
.withCredential(auth)
.withRegion(CbrRegion.valueOf("<YOUR REGION>"))
.build();
ListVaultRequest request = new ListVaultRequest();
request.withLimit(<limit>);
request.withName("<name>");
request.withOffset(<offset>);
request.withCloudType(ListVaultRequest.CloudTypeEnum.fromValue("<cloud_type>"));
request.withProtectType(ListVaultRequest.ProtectTypeEnum.fromValue("<protect_type>"));
request.withObjectType("<object_type>");
request.withEnterpriseProjectId("<enterprise_project_id>");
request.withId();
request.withPolicyId("<policy_id>");
request.withStatus("<status>");
request.withResourceIds("<resource_ids>");
try {
    ListVaultResponse response = client.listVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListVaultRequest()
        request.limit = <limit>
        request.name = "<name>"
        request.offset = <offset>
        request.cloud_type = "<cloud_type>"
        request.protect_type = "<protect_type>"
        request.object_type = "<object_type>"
        request.enterprise_project_id = "<enterprise_project_id>"
        request.id =
        request.policy_id = "<policy_id>"
        request.status = "<status>"
```

```
request.resource_ids = "<resource_ids>"  
response = client.list_vault(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListVaultRequest{}  
    limitRequest:= int32(<limit>)  
    request.Limit = &limitRequest  
    nameRequest:= "<name>"  
    request.Name = &nameRequest  
    offsetRequest:= int32(<offset>)  
    request.Offset = &offsetRequest  
    cloudTypeRequest:= model.GetListVaultRequestCloudTypeEnum().<CLOUD_TYPE>  
    request.CloudType = &cloudTypeRequest  
    protectTypeRequest:= model.GetListVaultRequestProtectTypeEnum().<PROTECT_TYPE>  
    request.ProtectType = &protectTypeRequest  
    objectTypeRequest:= "<object_type>"  
    request.ObjectType = &objectTypeRequest  
    enterpriseProjectIdRequest:= "<enterprise_project_id>"  
    request.EnterpriseProjectId = &enterpriseProjectIdRequest  
    policyIdRequest:= "<policy_id>"  
    request.PolicyId = &policyIdRequest  
    statusRequest:= "<status>"  
    request.Status = &statusRequest  
    resourceIdsRequest:= "<resource_ids>"  
    request.ResourceIds = &resourceIdsRequest  
    response, err := client.ListVault(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.4 修改存储库

功能介绍

根据存储库ID修改存储库

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/vaults/{vault_id}

表 4-199 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-200 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-201 请求 Body 参数

参数	是否必选	参数类型	描述
vault	是	VaultUpdate object	存储库修改参数体

表 4-202 VaultUpdate

参数	是否必选	参数类型	描述
billing	否	BillingUpdate object	账单信息
name	否	String	存储库名称 最小长度: 1 最大长度: 64
auto_bind	否	Boolean	是否支持自动挂载
bind_rules	否	VaultBindRules object	绑定规则
auto_expand	否	Boolean	是否开启存储库自动扩容能力 (只支持按需存储库)。
smn_notify	否	Boolean	发送smn通知开关 缺省值: true
threshold	否	Integer	存储库容量阈值, 存储库已用容量和总容量的百分比超过该值, 若smn_notify为开, 将发送相关通知。 最小值: 1 最大值: 100 缺省值: 80

表 4-203 BillingUpdate

参数	是否必选	参数类型	描述
consistent_level	否	String	存储库规格 枚举值: <ul style="list-style-type: none">• app_consistent• crash_consistent

参数	是否必选	参数类型	描述
size	否	Integer	存储库大小，单位为GB 最小值：1 最大值：10485760

表 4-204 VaultBindRules

参数	是否必选	参数类型	描述
tags	否	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度：0 最大长度：5 数组长度：0 - 5

表 4-205 BindRulesTags

参数	是否必选	参数类型	描述
key	是	String	key不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文，字母，数字， “-” , “_” 组成。
value	是	String	value不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文，字母，数字， “-” , “_” , “.” 组成。

响应参数

状态码： 200

表 4-206 响应 Body 参数

参数	参数类型	描述
vault	Vault object	存储库查询返回对象

表 4-207 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’ 。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80

参数	参数类型	描述
sys_lock_source_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值: <ul style="list-style-type: none">• SMB• "

表 4-208 Billing

参数	参数类型	描述
allocated	Integer	已分配容量, 单位GB
charging_mode	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal

参数	参数类型	描述
status	String	存储库状态 枚举值： <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-209 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小

参数	参数类型	描述
backup_count	Integer	副本数量

表 4-210 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-211 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可 选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-212 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-213 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

修改一个存储库，容量为200G，名称为vault_name02。

```
PUT https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d

{
  "vault": {
    "billing": {
      "size": 200
    },
    "name": "vault_name02"
  }
}
```

响应示例

状态码： 200

OK

```
{
  "vault": {
    "name": "vault_name02",
    "billing": {
      "status": "available",
      "used": 0,
      "protect_type": "backup",
      "object_type": "server",
      "allocated": 40,
      "spec_code": "vault.backup.server.normal",
    }
  }
}
```

```
        "size" : 200,
        "cloud_type" : "public",
        "consistent_level" : "crash_consistent",
        "charging_mode" : "post_paid"
    },
    "tags" : [ {
        "value" : "value01",
        "key" : "key01"
    }],
    "created_at" : "2019-05-08T09:31:21.840+00:00",
    "project_id" : "4229d7a45436489f8c3dc2b1d35d4987",
    "enterprise_project_id" : 0,
    "id" : "2402058d-8373-4b0a-b848-d3c0dfdc71a8",
    "resources" : [ {
        "name" : "ecs-1f0f-0003",
        "protect_status" : "available",
        "extra_info" : {
            "exclude_volumes" : [ "bdef09bb-293f-446a-88a4-86e9f14408c4" ]
        },
        "type" : "OS::Nova::Server",
        "id" : "07595625-198e-4e4d-879b-9d53f68ba551",
        "size" : 40
    }],
    "description" : "vault_description"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改一个存储库，容量为200G，名称为vault_name02。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class UpdateVaultSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateVaultRequest request = new UpdateVaultRequest();
```

```
VaultUpdateReq body = new VaultUpdateReq();
BillingUpdate billingVault = new BillingUpdate();
billingVault.withSize(200);
VaultUpdate vaultbody = new VaultUpdate();
vaultbody.withBilling(billingVault)
    .withName("vault_name02");
body.withVault(vaultbody);
request.withBody(body);
try {
    UpdateVaultResponse response = client.updateVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

修改一个存储库，容量为200G，名称为vault_name02。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateVaultRequest()
        billingVault = BillingUpdate(
            size=200
        )
        vaultbody = VaultUpdate(
            billing=billingVault,
            name="vault_name02"
        )
        request.body = VaultUpdateReq(
            vault=vaultbody
        )
        response = client.update_vault(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

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

Go

修改一个存储库，容量为200G，名称为vault_name02。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region" "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVaultRequest{}
    sizeBilling:= int32(200)
    billingVault := &model.BillingUpdate{
        Size: &sizeBilling,
    }
    nameVault:= "vault_name02"
    vaultbody := &model.VaultUpdate{
        Billing: billingVault,
        Name: &nameVault,
    }
    request.Body = &model.VaultUpdateReq{
        Vault: vaultbody,
    }
    response, err := client.UpdateVault(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.5 删除存储库

功能介绍

删除存储库。若删除储存库，将一并删除储存库中的所有备份。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/vaults/{vault_id}

表 4-214 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-215 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

无

请求示例

删除指定存储库。

```
DELETE https://{endpoint}/v3/{project_id}/vaults/{vault_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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DeleteVaultSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteVaultRequest request = new DeleteVaultRequest();
        try {
            DeleteVaultResponse response = client.deleteVault(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteVaultRequest{}
    response, err := client.DeleteVault(request)
```

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

更多

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

状态码

状态码	描述
204	No Content

错误码

请参见[错误码](#)。

4.5.6 移除资源

功能介绍

移除存储库中的资源，若移除资源，将一并删除该资源在保管库中的备份

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/{vault_id}/removeresources

表 4-216 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-217 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-218 请求 Body 参数

参数	是否必选	参数类型	描述
resource_ids	是	Array of strings	要移除的资源ID列表 最小长度：1 最大长度：256 数组长度：1 - 256

响应参数

状态码： 200

表 4-219 响应 Body 参数

参数	参数类型	描述
remove_resource_ids	Array of strings	移除的资源ID

请求示例

移除存储库中的指定资源，并删除该资源在存储库中的备份。

```
POST https://[endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d/removeresources
{
  "resource_ids" : [ "97595625-198e-4e4d-879b-9d53f68ba551" ]
}
```

响应示例

状态码： 200

OK

```
{
  "remove_resource_ids" : [ "97595625-198e-4e4d-879b-9d53f68ba551" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

移除存储库中的指定资源，并删除该资源在存储库中的备份。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class RemoveVaultResourceSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        RemoveVaultResourceRequest request = new RemoveVaultResourceRequest();
        VaultRemoveResourceReq body = new VaultRemoveResourceReq();
        List<String> listbodyResourcelds = new ArrayList<>();
        listbodyResourcelds.add("97595625-198e-4e4d-879b-9d53f68ba551");
        body.withResourcelds(listbodyResourcelds);
        request.withBody(body);
        try {
            RemoveVaultResourceResponse response = client.removeVaultResource(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

移除存储库中的指定资源，并删除该资源在存储库中的备份。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = RemoveVaultResourceRequest()
        listResourceldsbody = [
            "97595625-198e-4e4d-879b-9d53f68ba551"
        ]
        request.body = VaultRemoveResourceReq(
            resource_ids=listResourceldsbody
        )
        response = client.remove_vault_resource(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

移除存储库中的指定资源，并删除该资源在存储库中的备份。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).  
WithCredential(auth).  
Build()  
  
request := &model.RemoveVaultResourceRequest{  
var listResourceIdsbody = []string{  
    "97595625-198e-4e4d-879b-9d53f68ba551",  
}  
request.Body = &model.VaultRemoveResourceReq{  
    ResourceIds: listResourceIdsbody,  
}  
response, err := client.RemoveVaultResource(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.7 添加资源

功能介绍

存储库添加资源

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/{vault_id}/addresources

表 4-220 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-221 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-222 请求 Body 参数

参数	是否必选	参数类型	描述
resources	是	Array of ResourceCreate objects	资源列表 最小长度：1 最大长度：256 数组长度：1 - 256

表 4-223 ResourceCreate

参数	是否必选	参数类型	描述
extra_info	否	ResourceExtraInfo object	资源附加信息
id	是	String	待备份资源id
type	是	String	待备份资源的类型： OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
name	否	String	名称 最小长度：0 最大长度：255

表 4-224 ResourceExtraInfo

参数	是否必选	参数类型	描述
exclude_volumes	否	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有效，排除不需要备份的磁盘。当虚拟机新绑定磁盘时，也能继续排除之前设置不用备份的卷。

响应参数

状态码： 200

表 4-225 响应 Body 参数

参数	参数类型	描述
add_resource_ids	Array of strings	已添加的资源ID列表

请求示例

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d/addresources

{
  "resources": [
    {
      "extra_info": {
        "exclude_volumes": [
          "bdef09bb-293f-446a-88a4-86e9f14408c4"
        ]
      },
      "id": "97595625-198e-4e4d-879b-9d53f68ba551",
      "type": "OS::Nova::Server"
    }
  ]
}
```

响应示例

状态码： 200

OK

```
{
  "add_resource_ids": [
    "97595625-198e-4e4d-879b-9d53f68ba551"
]
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class AddVaultResourceSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        AddVaultResourceRequest request = new AddVaultResourceRequest();
        VaultAddResourceReq body = new VaultAddResourceReq();
        List<String> listExtraInfoExcludeVolumes = new ArrayList<>();
        listExtraInfoExcludeVolumes.add("bdef09bb-293f-446a-88a4-86e9f14408c4");
        ResourceExtraInfo extraInfoResources = new ResourceExtraInfo();
        extraInfoResources.withExcludeVolumes(listExtraInfoExcludeVolumes);
        List<ResourceCreate> listbodyResources = new ArrayList<>();
        listbodyResources.add(
            new ResourceCreate()
                .withExtraInfo(extraInfoResources)
                .withId("97595625-198e-4e4d-879b-9d53f68ba551")
                .withType("OS::Nova::Server")
        );
        body.withResources(listbodyResources);
        request.withBody(body);
        try {
            AddVaultResourceResponse response = client.addVaultResource(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddVaultResourceRequest()
        listExcludeVolumesExtraInfo = [
            "bdef09bb-293f-446a-88a4-86e9f14408c4"
        ]
        extraInfoResources = ResourceExtraInfo(
            exclude_volumes=listExcludeVolumesExtraInfo
        )
        listResourcesbody = [
            ResourceCreate(
                extra_info=extraInfoResources,
                id="97595625-198e-4e4d-879b-9d53f68ba551",
                type="OS::Nova::Server"
            )
        ]
        request.body = VaultAddResourceReq(
            resources=listResourcesbody
        )
        response = client.add_vault_resource(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddVaultResourceRequest{}
var listExcludeVolumesExtraInfo = []string{
    "bdef09bb-293f-446a-88a4-86e9f14408c4",
}
extraInfoResources := &model.ResourceExtraInfo{
    ExcludeVolumes: &listExcludeVolumesExtraInfo,
}
var listResourcesbody = []model.ResourceCreate{
{
    ExtraInfo: extraInfoResources,
    Id: "97595625-198e-4e4d-879b-9d53f68ba551",
    Type: "OS::Nova::Server",
},
}
request.Body = &model.VaultAddResourceReq{
    Resources: listResourcesbody,
}
response, err := client.AddVaultResource(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.8 设置存储库策略

功能介绍

存储库设置策略

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/{vault_id}/associatepolicy

表 4-226 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-227 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-228 请求 Body 参数

参数	是否必选	参数类型	描述
destination_vault_id	否	String	目标vault ID，只有设置复制策略时使用，而且必传
policy_id	否	String	策略ID。policy_id字段与add_policy_ids字段在一次请求中有且只有一个。
add_policy_ids	否	Array of strings	多策略场景下，绑定新策略的id列表。policy_id字段与add_policy_ids字段在一次请求中有且只有一个。 最小长度：1 最大长度：10

响应参数

状态码： 200

表 4-229 响应 Body 参数

参数	参数类型	描述
associate_policy	VaultPolicyResp object	绑定策略详情

表 4-230 VaultPolicyResp

参数	参数类型	描述
destination_vault_id	String	目标region的vault ID，仅设置复制策略时有。
policy_id	String	设置的策略ID
vault_id	String	设置策略的vault ID

请求示例

设置存储库的备份策略。

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d/associatepolicy
{
    "policy_id" : "7075c397-25a0-43e2-a83a-bb16eaca3ee5"
}
```

响应示例

状态码： 200

OK

```
{
    "associate_policy" : {
        "vault_id" : "2402058d-8373-4b0a-b848-d3c0dfdc71a8",
        "policy_id" : "7075c397-25a0-43e2-a83a-bb16eaca3ee5"
    }
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置存储库的备份策略。

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

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

```
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class AssociateVaultPolicySolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        AssociateVaultPolicyRequest request = new AssociateVaultPolicyRequest();
        VaultAssociate body = new VaultAssociate();
        body.withPolicyId("7075c397-25a0-43e2-a83a-bb16eaca3ee5");
        request.withBody(body);
        try {
            AssociateVaultPolicyResponse response = client.associateVaultPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

设置存储库的备份策略。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.v1 import *

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

    credentials = BasicCredentials(ak, sk) \
        client = CbrClient.new_builder() \
            .with_credentials(credentials) \
```

```
.with_region(CbrRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = AssociateVaultPolicyRequest()  
    request.body = VaultAssociate(  
        policy_id="7075c397-25a0-43e2-a83a-bb16eaca3ee5"  
    )  
    response = client.associate_vault_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

设置存储库的备份策略。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
    request := &model.AssociateVaultPolicyRequest{}  
    policyIdVaultAssociate:= "7075c397-25a0-43e2-a83a-bb16eaca3ee5"  
    request.Body = &model.VaultAssociate{  
        PolicyId: &policyIdVaultAssociate,  
    }  
    response, err := client.AssociateVaultPolicy(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.9 解除存储库策略

功能介绍

存储库解除策略

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/{vault_id}/dissociatepolicy

表 4-231 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-232 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-233 请求 Body 参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略ID

响应参数

状态码： 200

表 4-234 响应 Body 参数

参数	参数类型	描述
dissociate_policy	VaultPolicyResp object	存储库解绑策略详情

表 4-235 VaultPolicyResp

参数	参数类型	描述
destination_vault_id	String	目标region的vault ID，仅设置复制策略时有。
policy_id	String	设置的策略ID
vault_id	String	设置策略的vault ID

请求示例

解除存储库的备份策略。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d/dissociatepolicy
{
    "policy_id" : "7075c397-25a0-43e2-a83a-bb16eaca3ee5"
}
```

响应示例

状态码： 200

OK

```
{
    "dissociate_policy" : {
        "vault_id" : "2402058d-8373-4b0a-b848-d3c0dfdc71a8",
        "policy_id" : "7075c397-25a0-43e2-a83a-bb16eaca3ee5"
    }
}
```

SDK 代码示例

SDK代码示例如下。

Java

解除存储库的备份策略。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DisassociateVaultPolicySolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        DisassociateVaultPolicyRequest request = new DisassociateVaultPolicyRequest();
        VaultDissociate body = new VaultDissociate();
        body.withPolicyId("7075c397-25a0-43e2-a83a-bb16eaca3ee5");
        request.withBody(body);
        try {
            DisassociateVaultPolicyResponse response = client.disassociateVaultPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

解除存储库的备份策略。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DisassociateVaultPolicyRequest()
        request.body = VaultDissociate(
            policy_id="7075c397-25a0-43e2-a83a-bb16eaca3ee5"
        )
        response = client.disassociate_vault_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

解除存储库的备份策略。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisassociateVaultPolicyRequest{}
    request.Body = &model.VaultDissociate{
```

```
    PolicyId: "7075c397-25a0-43e2-a83a-bb16eaca3ee5",
}
response, err := client.DisassociateVaultPolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.10 查询其他区域存储库列表

功能介绍

查询其他区域的存储库列表

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vaults/external

表 4-236 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-237 Query 参数

参数	是否必选	参数类型	描述
cloud_type	否	String	云类型。取值为public和hybrid。

参数	是否必选	参数类型	描述
external_project_id	是	String	其他区域的项目ID
limit	否	Integer	每页显示条目数 最小值: 1 最大值: 1000
object_type	否	String	资源类型
offset	否	Integer	偏移值
protect_type	否	String	保护类型。取值为backup, replication和hybrid。
region_id	是	String	区域ID
vault_id	否	String	存储库ID，指定存储ID时其他过滤条件不生效。

请求参数

无

响应参数

状态码: 200

表 4-238 响应 Body 参数

参数	参数类型	描述
vaults	Array of Vault objects	存储库实例列表
count	Integer	存储库个数
limit	Integer	每页显示的条目数量 最小值: 1 最大值: 1000 缺省值: 1000
offset	Integer	偏移量，表示从此偏移量开始查询 最小值: 0 缺省值: 0

表 4-239 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’ 。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值: 1 最大值: 100 缺省值: 80

参数	参数类型	描述
sys_lock_source_service	String	用于标识SMB服务 最小长度: 0 最大长度: 32 枚举值: <ul style="list-style-type: none">• SMB• "

表 4-240 Billing

参数	参数类型	描述
allocated	Integer	已分配容量, 单位GB
charging_mode	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型: 备份 (backup)、复制 (replication)。
size	Integer	容量, 单位GB 最小值: 1 最大值: 10485760
spec_code	String	规格编码。云服务备份存储库: vault.backup.server.normal; 云硬盘备份存储库: vault.backup.volume.normal; 文件备份存储库: vault.backup.turbo.normal

参数	参数类型	描述
status	String	存储库状态 枚举值： <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-241 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小

参数	参数类型	描述
backup_count	Integer	副本数量

表 4-242 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-243 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 key只能由中文, 字母, 数字, “-”, “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可 选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 value只能由中文, 字母, 数字, “-”, “_”, “.” 组成。

表 4-244 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度：0 最大长度：5 数组长度：0 - 5

表 4-245 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

请求示例

```
GET /v3/{project_id}/vaults/external?  
external_project_id=68589cac08274b82b4e254268a3862d8&region_id=cn-shenzhen-1
```

响应示例

状态码： 200

OK

```
{  
    "count": 1,  
    "vaults": [ {  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "name": "vault-name",  
        "billing": {  
            "status": "available",  
            "used": 5588,  
            "protect_type": "replication",  
            "object_type": "server",  
            "consistent_level": "crash_consistent",  
            "cloud_type": "public",  
            "allocated": 0,  
            "charging_mode": "post_paid",  
            "spec_code": "vault.replication.server.normal",  
            "size": 10  
        },  
        "tags": [ ],  
        "created_at": "2019-04-08T11:06:25.766+00:00",  
        "enterprise_project_id": "876ae48d-cfb3-4aff-bb55-83167abd3876",  
        "project_id": "68589cac08274b82b4e254268a3862d8",  
        "id": "0ca3eb86-8800-46da-9c37-9d657a825274",  
    }]
```

```
        "resources" : [ ]  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ListExternalVaultSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListExternalVaultRequest request = new ListExternalVaultRequest();  
        request.withExternalProjectId("<external_project_id>");  
        request.withLimit(<limit>);  
        request.withOffset(<offset>);  
        request.withProtectType(ListExternalVaultRequest.ProtectTypeEnum.fromValue("<protect_type>"));  
        request.withRegionId("<region_id>");  
        request.withObjcetType("<objcet_type>");  
        request.withCloudType("<cloud_type>");  
        request.withVaultId("<vault_id>");  
        try {  
            ListExternalVaultResponse response = client.listExternalVault(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListExternalVaultRequest()
        request.external_project_id = "<external_project_id>"
        request.limit = <limit>
        request.offset = <offset>
        request.protect_type = "<protect_type>"
        request.region_id = "<region_id>"
        request.object_type = "<object_type>"
        request.cloud_type = "<cloud_type>"
        request.vault_id = "<vault_id>"
        response = client.list_external_vault(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListExternalVaultRequest{}
request.ExternalProjectId = "<external_project_id>"
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
protectTypeRequest:= model.GetListExternalVaultRequestProtectTypeEnum().<PROTECT_TYPE>
request.ProtectType = &protectTypeRequest
request.RegionId = "<region_id>"
objecTypeRequest:= "<objecet_type>"
request.ObjecType = &objecTypeRequest
cloudTypeRequest:= "<cloud_type>"
request.CloudType = &cloudTypeRequest
vaultIdRequest:= "<vault_id>"
request.VaultId = &vaultIdRequest
response, err := client.ListExternalVault(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.11 迁移资源

功能介绍

支持资源迁移到另一个存储库，不删除备份。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/{vault_id}/migrateresources

表 4-246 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vault_id	是	String	存储库ID

请求参数

表 4-247 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-248 请求 Body 参数

参数	是否必选	参数类型	描述
destination_vault_id	是	String	目标存储库
resource_ids	是	Array of strings	待迁移的资源ID

响应参数

状态码： 200

表 4-249 响应 Body 参数

参数	参数类型	描述
migrated_resources	Array of strings	迁移资源列表

请求示例

迁移备份资源从源存储库到目标存储库且不删除源备份。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/79bd9daa-884f-4f84-b8fe-235d58cd927d/migrateresources
{
    "resource_ids" : [ "abcdde3f-e0e3-403a-b690-fc259dd70008" ],
```

```
        "destination_vault_id" : "fe578a6c-d1a8-4790-bd52-5954af4d446c"  
    }
```

响应示例

状态码： 200

OK

```
{  
    "migrated_resources" : [ "fe578a6c-d1a8-4790-bd52-5954af4d446c" ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

迁移备份资源从源存储库到目标存储库且不删除源备份。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class MigrateVaultResourceSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        MigrateVaultResourceRequest request = new MigrateVaultResourceRequest();  
        VaultMigrateResourceReq body = new VaultMigrateResourceReq();  
        List<String> listbodyResourcelds = new ArrayList<>();  
        listbodyResourcelds.add("abcdde3f-e0e3-403a-be90-fc259dd70008");  
        body.withResourcelds(listbodyResourcelds);  
        body.withDestinationVaultId("fe578a6c-d1a8-4790-bd52-5954af4d446c");  
        request.withBody(body);  
        try {  
            MigrateVaultResourceResponse response = client.migrateVaultResource(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

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

Python

迁移备份资源从源存储库到目标存储库且不删除源备份。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = MigrateVaultResourceRequest()
        listResourceldsbody = [
            "abcdde3f-e0e3-403a-b690-fc259dd70008"
        ]
        request.body = VaultMigrateResourceReq(
            resource_ids=listResourceldsbody,
            destination_vault_id="fe578a6c-d1a8-4790-bd52-5954af4d446c"
        )
        response = client.migrate_vault_resource(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

迁移备份资源从源存储库到目标存储库且不删除源备份。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.MigrateVaultResourceRequest{  
        var listResourceIdsBody = []string{  
            "abcdde3f-e0e3-403a-b690-fc259dd70008",  
        }  
        request.Body = &model.VaultMigrateResourceReq{  
            ResourceIds: listResourceIdsBody,  
            DestinationVaultId: "fe578a6c-d1a8-4790-bd52-5954af4d446c",  
        }  
        response, err := client.MigrateVaultResource(request)  
        if err == nil {  
            fmt.Printf("%+v\n", response)  
        } else {  
            fmt.Println(err)  
        }  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.12 批量修改存储库

功能介绍

批量修改项目下所有存储库

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/vaults/batch-update

表 4-250 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-251 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-252 请求 Body 参数

参数	是否必选	参数类型	描述
vault	是	VaultBatchUpdate object	存储库修改参数体

表 4-253 VaultBatchUpdate

参数	是否必选	参数类型	描述
smn_notify	否	Boolean	存储库smn消息通知开关 缺省值：true
threshold	否	Integer	存储库容量阈值 最小值：1 最大值：100 缺省值：80

响应参数

状态码： 200

表 4-254 响应 Body 参数

参数	参数类型	描述
updated_vaults_id	Array of strings	已批量修改id列表 最小长度： 0

请求示例

批量修改所有存储库，容量阈值为60G。

```
PUT https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/batch-update
{
  "vault" : {
    "smn_notify" : true,
    "threshold" : 60
  }
}
```

响应示例

状态码： 200

OK

```
{
  "updated_vaults_id" : [ "79bd9daa-884f-4f84-b8fe-235d58cd927d" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量修改所有存储库，容量阈值为60G。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class BatchUpdateVaultSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
BatchUpdateVaultRequest request = new BatchUpdateVaultRequest();
BatchUpdateVaultRequestBody body = new BatchUpdateVaultRequestBody();
VaultBatchUpdate vaultbody = new VaultBatchUpdate();
vaultbody.withSmnNotify(true)
    .withThreshold(60);
body.withVault(vaultbody);
request.withBody(body);
try {
    BatchUpdateVaultResponse response = client.batchUpdateVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

批量修改所有存储库，容量阈值为60G。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchUpdateVaultRequest()
        vaultbody = VaultBatchUpdate(
            smn_notify=True,
            threshold=60
    
```

```
)  
    request.body = BatchUpdateVaultRequestBody(  
        vault=vaultbody  
)  
    response = client.batch_update_vault(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

批量修改所有存储库，容量阈值为60G。

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
    request := &model.BatchUpdateVaultRequest{  
        smnNotifyVault:= true  
        thresholdVault:= int32(60)  
        vaultbody := &model.VaultBatchUpdate{  
            SmnNotify: &smnNotifyVault,  
            Threshold: &thresholdVault,  
        }  
        request.Body = &model.BatchUpdateVaultRequestBody{  
            Vault: vaultbody,  
        }  
    response, err := client.BatchUpdateVault(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.13 设置存储库资源

功能介绍

设置存储库资源是否自动备份

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/vaults/{vault_id}/set-resources

表 4-255 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
vault_id	是	String	存储库id

请求参数

表 4-256 请求 Body 参数

参数	是否必选	参数类型	描述
resource_ids	是	Array of strings	需要设置的资源id。
action	是	String	设置存储库资源动作 枚举值： • suspend • unsuspend

响应参数

状态码： 200

表 4-257 响应 Body 参数

参数	参数类型	描述
set_resource_ids	Array of strings	本次设置的资源id列表。

请求示例

设置存储库资源自动备份

```
PUT https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/2b076f26-391f-40fa-bf71-a798940faccf/set_resource

{
  "resource_ids" : [ "2b076f26-391f-40fa-bf71-a798940facbb", "dfx76f26-391f-40fa-bf71-a798940faca" ],
  "action" : "suspend"
}
```

响应示例

状态码： 200

OK

```
{
  "set_resource_ids" : [ "2b076f26-391f-40fa-bf71-a798940facbb", "dfx76f26-391f-40fa-bf71-a798940faca" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

设置存储库资源自动备份

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class SetVaultResourceSolution {

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

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

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_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);

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

SetVaultResourceRequest request = new SetVaultResourceRequest();
VaultSetResourceReq body = new VaultSetResourceReq();
List<String> listbodyResourcelds = new ArrayList<>();
listbodyResourcelds.add("2b076f26-391f-40fa-bf71-a798940facbb");
listbodyResourcelds.add("dfx76f26-391f-40fa-bf71-a798940faca");
body.withAction(VaultSetResourceReq.ActionEnum.fromValue("suspend"));
body.withResourcelds(listbodyResourcelds);
request.withBody(body);
try {
    SetVaultResourceResponse response = client.setVaultResource(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

设置存储库资源自动备份

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \\\
        .with_credentials(credentials) \\
        .with_region(CbrRegion.value_of("<YOUR REGION>")) \\
        .build()

    try:
        request = SetVaultResourceRequest()
```

```
listResourceIdsbody = [
    "2b076f26-391f-40fa-bf71-a798940facbb",
    "dfx76f26-391f-40fa-bf71-a798940facaa"
]
request.body = VaultSetResourceReq(
    action="suspend",
    resource_ids=listResourceIdsbody
)
response = client.set_vault_resource(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

设置存储库资源自动备份

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SetVaultResourceRequest{}
    var listResourceIdsbody = []string{
        "2b076f26-391f-40fa-bf71-a798940facbb",
        "dfx76f26-391f-40fa-bf71-a798940facaa",
    }
    request.Body = &model.VaultSetResourceReq{
        Action: model.GetVaultSetResourceReqActionEnum().SUSPEND,
        ResourceIds: listResourceIdsbody,
    }
    response, err := client.SetVaultResource(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.5.14 创建包周期存储库

功能介绍

创建包周期存储库

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vaults/order

表 4-258 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-259 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-260 请求 Body 参数

参数	是否必选	参数类型	描述
vault	是	VaultOrder object	存储库

表 4-261 VaultOrder

参数	是否必选	参数类型	描述
name	否	String	存储库名称 最小长度：1 最大长度：64 最小长度：1 最大长度：64
billing	是	BillingCreate object	创建参数信息
resources	是	Array of ResourceCreate objects	绑定的备份资源，未在创建时绑定资源填[] 数组长度：0 - 256
description	否	String	描述 最小长度：0 最大长度：255 最小长度：0 最大长度：255
backup_policy_id	否	String	备份策略ID，不设置时为null，不自动备份。
tags	否	Array of Tag objects	标签列表 tags不允许为空列表。tags中最多包含10个key。tags中key不允许重复。 数组长度：1 - 10
enterprise_project_id	否	String	企业项目ID，默认为‘0’。
auto_bind	否	Boolean	是否支持自动挂载。
bind_rules	否	VaultBindRules object	自动挂载的规则
threshold	否	Integer	存储库阈值，百分比。 最小值：1 最大值：100
smn_notify	否	Boolean	当容量到达阈值，是否启用通知

参数	是否必选	参数类型	描述
parameters	否	VaultCreateParameters object	存储库订单联合参数
auto_expand	否	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。

表 4-262 BillingCreate

参数	是否必选	参数类型	描述
cloud_type	否	String	公有云:public ;混合云:hybrid
consistent_level	是	String	规格, 崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	是	String	对象类型: 云服务器 (server), 云硬盘 (disk), 文件系统 (turbo), 云桌面 (workspace), VMware (vmware), 关系型数据库 (rds), 文件 (file)。
protect_type	是	String	保护类型: 备份 (backup)、复制(replication)。
size	是	Integer	容量, 单位GB 最小值: 10 最大值: 10485760
charging_mode	否	String	创建模式, 按需: post_paid, 包周期: pre_paid, 默认为 post_paid 缺省值: post_paid
period_type	否	String	创建类型, charging_mode为 pre_paid 必填, 按年(year)或者按月(month) 枚举值: <ul style="list-style-type: none">• year• month
period_num	否	Integer	创建类型的数量, charging_mode为 pre_paid 必填
is_auto_renew	否	Boolean	到期后是否自动续期, 默认不续期 缺省值: false

参数	是否必选	参数类型	描述
is_auto_pay	否	Boolean	是否自动付费， 默认为不自动付费 缺省值： false
console_url	否	String	跳转URL 最小长度： 1 最大长度： 255
is_multi_az	否	Boolean	存储库多az属性， 默认为false 缺省值： false

表 4-263 ResourceCreate

参数	是否必选	参数类型	描述
extra_info	否	ResourceExtraInfo object	资源附加信息
id	是	String	待备份资源id
type	是	String	待备份资源的类型： OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
name	否	String	名称 最小长度： 0 最大长度： 255

表 4-264 ResourceExtraInfo

参数	是否必选	参数类型	描述
exclude_volumes	否	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有效，排除不需要备份的磁盘。当虚拟机新绑定磁盘时，也能继续排除之前设置不用备份的卷。

表 4-265 Tag

参数	是否必选	参数类型	描述
key	是	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	是	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-266 VaultBindRules

参数	是否必选	参数类型	描述
tags	否	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度: 0 最大长度: 5 数组长度: 0 - 5

表 4-267 BindRulesTags

参数	是否必选	参数类型	描述
key	是	String	key不能包含非打印字符 ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 key只能由中文, 字母, 数字, “-”, “_”组成。
value	是	String	value不能包含非打印字符 ASCII(0-31), “=”, “*”, “<”, “>”, “\”, “,”, “ ”, “/”。 value只能由中文, 字母, 数字, “-”, “_”, “.”组成。

表 4-268 VaultCreateParameters

参数	是否必选	参数类型	描述
combined_order	否	CombinedOrder object	组合订单。

表 4-269 CombinedOrder

参数	是否必选	参数类型	描述
combined_order_id	否	String	组合订单 ID
combined_order_ecs_num	否	Integer	组合订单中 ECS 服务器数量，当前批量最大为 500。 最小值: 1 最大值: 1000 最小值: 1 最大值: 1000
combined_order_num	否	Integer	组合订单数量。 最小值: 1 最大值: 1000 最小值: 1 最大值: 1000

响应参数

状态码： 200

表 4-270 响应 Body 参数

参数	参数类型	描述
orders	Array of CbcOrderResult objects	订单详情
retCode	Integer	创建结果代码 0: 成功
errText	String	创建结果信息
error_code	String	操作错误码 0: 无错误

表 4-271 CbcOrderResult

参数	参数类型	描述
cloudServiceId	String	云服务ID
orderId	String	订单ID
subscribeResult	Integer	订购结果，1: 成功；0: 失败
resourceId	String	包周期资源预生成资源id。

状态码： 400

表 4-272 响应 Body 参数

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

请求示例

- 创建一个包周期云服务器备份存储库，存储库容量为80G，周期为一个月且绑定一个资源。

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/order
{
    "vault": {
        "name": "vault_name",
        "billing": {
            "consistent_level": "app_consistent",
```

```
        "object_type" : "server",
        "protect_type" : "backup",
        "size" : 80,
        "charging_mode" : "pre_paid",
        "period_type" : "month",
        "period_num" : 1
    },
    "resources" : [ {
        "id" : "23a320a5-3efd-4568-b1aa-8dd9183cc64c",
        "type" : "OS::Nova::Server"
    } ]
}
```

- 创建一个包周期云服务器备份存储库，存储库容量阈值为80G，云服务器数量为5，周期为一个月同时绑定一个资源并添加标签。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/vaults/order
```

```
{
    "vault" : {
        "name" : "123",
        "description" : "description",
        "backup_policy_id" : "70e9c916-7109-472d-94e7-120900e4ba61",
        "tags" : [ {
            "key" : "key",
            "value" : "value"
        }],
        "auto_bind" : false,
        "threshold" : 80,
        "smn_notify" : true,
        "parameters" : {
            "combined_order" : {
                "combined_order_id" : "CS2303030959MR1IT",
                "combined_order_ecs_num" : 5,
                "combined_order_num" : 5
            }
        },
        "bind_rules" : {
            "tags" : [ {
                "key" : "bind_rules_key",
                "value" : "bind_rules_value"
            }]
        },
        "billing" : {
            "cloud_type" : "public",
            "is_auto_renew" : false,
            "is_auto_pay" : false,
            "console_url" : "https://www.com",
            "consistent_level" : "app_consistent",
            "object_type" : "server",
            "protect_type" : "backup",
            "size" : 80,
            "charging_mode" : "pre_paid",
            "period_type" : "month",
            "period_num" : 1
        },
        "resources" : [ {
            "id" : "8ef08ba1-81f7-4e41-ae49-d3e2fb0ba388",
            "type" : "OS::Nova::Server",
            "name" : "server_name",
            "extra_info" : {
                "exclude_volumes" : [ "1855eb9a-2b5e-4938-a9f0-aea08b6f9243" ]
            }
        }]
    }
}
```

响应示例

状态码： 200

OK

```
{  
    "orders" : [ {  
        "orderId" : "CS2303030941NJJIW",  
        "cloudServiceId" : "cbr",  
        "subscribeResult" : 1,  
        "resourceId" : "ef3bf6bb-1ffb-46c0-8b23-f935ded5024a"  
    } ],  
    "retCode" : 0,  
    "errText" : "success",  
    "error_code" : 0  
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建一个包周期云服务器备份存储库，存储库容量为80G，周期为一个月且绑定一个资源。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreatePostPaidVaultSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreatePostPaidVaultRequest request = new CreatePostPaidVaultRequest();  
        VaultOrderCreateReqs body = new VaultOrderCreateReqs();  
        List<ResourceCreate> listVaultResources = new ArrayList<>();  
        listVaultResources.add(  
            new ResourceCreate()  
                .withId("23a320a5-3efd-4568-b1aa-8dd9183cc64c")
```

```
        .withType("OS::Nova::Server")
    );
    BillingCreate billingVault = new BillingCreate();
    billingVault.withConsistentLevel(BillingCreate.ConsistentLevelEnum.fromValue("app_consistent"))
        .withObjectType(BillingCreate.ObjectTypeEnum.fromValue("server"))
        .withProtectType(BillingCreate.ProtectTypeEnum.fromValue("backup"))
        .withSize(80)
        .withChargingMode(BillingCreate.ChargingModeEnum.fromValue("pre_paid"))
        .withPeriodType(BillingCreate.PeriodTypeEnum.fromValue("month"))
        .withPeriodNum(1);
    VaultOrder vaultbody = new VaultOrder();
    vaultbody.withName("vault_name")
        .withBilling(billingVault)
        .withResources(listVaultResources);
    body.withVault(vaultbody);
    request.withBody(body);
    try {
        CreatePostPaidVaultResponse response = client.createPostPaidVault(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- 创建一个包周期云服务器备份存储库，存储库容量阈值为80G，云服务器数量为5，周期为一个月同时绑定一个资源并添加标签。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class CreatePostPaidVaultSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
    }
}
```

```
CreatePostPaidVaultRequest request = new CreatePostPaidVaultRequest();
VaultOrderCreateReqs body = new VaultOrderCreateReqs();
CombinedOrder combinedOrderParameters = new CombinedOrder();
combinedOrderParameters.withCombinedOrderId("CS2303030959MR1IT")
    .withCombinedOrderEcsNum(5)
    .withCombinedOrderNum(5);
VaultCreateParameters parametersVault = new VaultCreateParameters();
parametersVault.withCombinedOrder(combinedOrderParameters);
List<BindRulesTags> listBindRulesTags = new ArrayList<>();
listBindRulesTags.add(
    new BindRulesTags()
        .withKey("bind_rules_key")
        .withValue("bind_rules_value")
);
VaultBindRules bindRulesVault = new VaultBindRules();
bindRulesVault.withTags(listBindRulesTags);
List<Tag> listVaultTags = new ArrayList<>();
listVaultTags.add(
    new Tag()
        .withKey("key")
        .withValue("value")
);
List<String> listExtraInfoExcludeVolumes = new ArrayList<>();
listExtraInfoExcludeVolumes.add("1855eb9a-2b5e-4938-a9f0-aea08b6f9243");
ResourceExtraInfo extraInfoResources = new ResourceExtraInfo();
extraInfoResources.withExcludeVolumes(listExtraInfoExcludeVolumes);
List<ResourceCreate> listVaultResources = new ArrayList<>();
listVaultResources.add(
    new ResourceCreate()
        .withExtraInfo(extraInfoResources)
        .withId("8ef08ba1-81f7-4e41-ae49-d3e2fb0ba388")
        .withType("OS::Nova::Server")
        .withName("server_name")
);
BillingCreate billingVault = new BillingCreate();
billingVault.withCloudType(BillingCreate.CloudTypeEnum.fromValue("public"))
    .withConsistentLevel(BillingCreate.ConsistentLevelEnum.fromValue("app_consistent"))
    .withObjectType(BillingCreate.ObjectTypeEnum.fromValue("server"))
    .withProtectType(BillingCreate.ProtectTypeEnum.fromValue("backup"))
    .withSize(80)
    .withChargingMode(BillingCreate.ChargingModeEnum.fromValue("pre_paid"))
    .withPeriodType(BillingCreate.PeriodTypeEnum.fromValue("month"))
    .withPeriodNum(1)
    .withIsAutoRenew(false)
    .withIsAutoPay(false)
    .withConsoleUrl("https://www.com");
VaultOrder vaultbody = new VaultOrder();
vaultbody.withName("123")
    .withBilling(billingVault)
    .withResources(listVaultResources)
    .withDescription("description")
    .withBackupPolicyId("70e9c916-7109-472d-94e7-120900e4ba61")
    .withTags(listVaultTags)
    .withAutoBind(false)
    .withBindRules(bindRulesVault)
    .withThreshold(80)
    .withSmnNotify(true)
    .withParameters(parametersVault);
body.withVault(vaultbody);
request.withBody(body);
try {
    CreatePostPaidVaultResponse response = client.createPostPaidVault(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
}
```

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

Python

- 创建一个包周期云服务器备份存储库，存储库容量为80G，周期为一个月且绑定一个资源。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreatePostPaidVaultRequest()
        listResourcesVault = [
            ResourceCreate(
                id="23a320a5-3efd-4568-b1aa-8dd9183cc64c",
                type="OS::Nova::Server"
            )
        ]
        billingVault = BillingCreate(
            consistent_level="app_consistent",
            object_type="server",
            protect_type="backup",
            size=80,
            charging_mode="pre_paid",
            period_type="month",
            period_num=1
        )
        vaultbody = VaultOrder(
            name="vault_name",
            billing=billingVault,
            resources=listResourcesVault
        )
        request.body = VaultOrderCreateReqs(
            vault=vaultbody
        )
        response = client.create_post_paid_vault(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 创建一个包周期云服务器备份存储库，存储库容量阈值为80G，云服务器数量为5，周期为一个月同时绑定一个资源并添加标签。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

try:
    request = CreatePostPaidVaultRequest()
    combinedOrderParameters = CombinedOrder(
        combined_order_id="CS2303030959MR1IT",
        combined_order_ecs_num=5,
        combined_order_num=5
    )
    parametersVault = VaultCreateParameters(
        combined_order=combinedOrderParameters
    )
    listTagsBindRules = [
        BindRulesTags(
            key="bind_rules_key",
            value="bind_rules_value"
        )
    ]
    bindRulesVault = VaultBindRules(
        tags=listTagsBindRules
    )
    listTagsVault = [
        Tag(
            key="key",
            value="value"
        )
    ]
    listExcludeVolumesExtraInfo = [
        "1855eb9a-2b5e-4938-a9f0-aea08b6f9243"
    ]
    extraInfoResources = ResourceExtraInfo(
        exclude_volumes=listExcludeVolumesExtraInfo
    )
    listResourcesVault = [
        ResourceCreate(
            extra_info=extraInfoResources,
            id="8ef08ba1-81f7-4e41-ae49-d3e2fb0ba388",
            type="OS::Nova::Server",
            name="server_name"
        )
    ]
    billingVault = BillingCreate(
        cloud_type="public",
        consistent_level="app_consistent",
        object_type="server",
    )
```

```
        protect_type="backup",
        size=80,
        charging_mode="pre_paid",
        period_type="month",
        period_num=1,
        is_auto_renew=False,
        is_auto_pay=False,
        console_url="https://www.com"
    )
    vaultbody = VaultOrder(
        name="123",
        billing=billingVault,
        resources=listResourcesVault,
        description="description",
        backup_policy_id="70e9c916-7109-472d-94e7-120900e4ba61",
        tags=listTagsVault,
        auto_bind=False,
        bind_rules=bindRulesVault,
        threshold=80,
        smn_notify=True,
        parameters=parametersVault
    )
    request.body = VaultOrderCreateReqs(
        vault=vaultbody
    )
    response = client.create_post_paid_vault(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- 创建一个包周期云服务器备份存储库，存储库容量为80G，周期为一个月且绑定一个资源。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
}
```

```
request := &model.CreatePostPaidVaultRequest{}
var listResourcesVault = []model.ResourceCreate{
    {
        Id: "23a320a5-3efd-4568-b1aa-8dd9183cc64c",
        Type: "OS::Nova::Server",
    },
}
chargingModeBilling:= model.GetBillingCreateChargingModeEnum().PRE_PAID
periodTypeBilling:= model.GetBillingCreatePeriodTypeEnum().MONTH
periodNumBilling:= int32(1)
billingVault := &model.BillingCreate{
    ConsistentLevel: model.GetBillingCreateConsistentLevelEnum().APP_CONSISTENT,
    ObjectType: model.GetBillingCreateObjectTypeEnum().SERVER,
    ProtectType: model.GetBillingCreateProtectTypeEnum().BACKUP,
    Size: int32(80),
    ChargingMode: &chargingModeBilling,
    PeriodType: &periodTypeBilling,
    PeriodNum: &periodNumBilling,
}
nameVault:= "vault_name"
vaultbody := &model.VaultOrder{
    Name: &nameVault,
    Billing: billingVault,
    Resources: listResourcesVault,
}
request.Body = &model.VaultOrderCreateReq{
    Vault: vaultbody,
}
response, err := client.CreatePostPaidVault(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

- 创建一个包周期云服务器备份存储库，存储库容量阈值为80G，云服务器数量为5，周期为一个月同时绑定一个资源并添加标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build())
    )
}
```

```
request := &model.CreatePostPaidVaultRequest{}
combinedOrderIdCombinedOrder:= "CS2303030959MR1IT"
combinedOrderEcsNumCombinedOrder:= int32(5)
combinedOrderNumCombinedOrder:= int32(5)
combinedOrderParameters := &model.CombinedOrder{
    CombinedOrderId: &combinedOrderIdCombinedOrder,
    CombinedOrderEcsNum: &combinedOrderEcsNumCombinedOrder,
    CombinedOrderNum: &combinedOrderNumCombinedOrder,
}
parametersVault := &model.VaultCreateParameters{
    CombinedOrder: combinedOrderParameters,
}
var listTagsBindRules = []model.BindRulesTags{
{
    Key: "bind_rules_key",
    Value: "bind_rules_value",
},
}
bindRulesVault := &model.VaultBindRules{
    Tags: &listTagsBindRules,
}
var listTagsVault = []model.Tag{
{
    Key: "key",
    Value: "value",
},
}
var listExcludeVolumesExtraInfo = []string{
    "1855eb9a-2b5e-4938-a9f0-aea08b6f9243",
}
extraInfoResources := &model.ResourceExtraInfo{
    ExcludeVolumes: &listExcludeVolumesExtraInfo,
}
nameResources:= "server_name"
var listResourcesVault = []model.ResourceCreate{
{
    ExtraInfo: extraInfoResources,
    Id: "8ef08ba1-81f7-4e41-ae49-d3e2fb0ba388",
    Type: "OS::Nova::Server",
    Name: &nameResources,
},
}
cloudTypeBilling:= model.GetBillingCreateCloudTypeEnum().PUBLIC
chargingModeBilling:= model.GetBillingCreateChargingModeEnum().PRE_PAID
periodTypeBilling:= model.GetBillingCreatePeriodTypeEnum().MONTH
periodNumBilling:= int32(1)
isAutoRenewBilling:= false
isAutoPayBilling:= false
consoleUrlBilling:= "https://www.com"
billingVault := &model.BillingCreate{
    CloudType: &cloudTypeBilling,
    ConsistentLevel: model.GetBillingCreateConsistentLevelEnum().APP_CONSISTENT,
    ObjectType: model.GetBillingCreateObjectTypeEnum().SERVER,
    ProtectType: model.GetBillingCreateProtectTypeEnum().BACKUP,
    Size: int32(80),
    ChargingMode: &chargingModeBilling,
    PeriodType: &periodTypeBilling,
    PeriodNum: &periodNumBilling,
    IsAutoRenew: &isAutoRenewBilling,
    IsAutoPay: &isAutoPayBilling,
    ConsoleUrl: &consoleUrlBilling,
}
nameVault:= "123"
descriptionVault:= "description"
backupPolicyIdVault:= "70e9c916-7109-472d-94e7-120900e4ba61"
autoBindVault:= false
thresholdVault:= int32(80)
smnNotifyVault:= true
vaultbody := &model.VaultOrder{
```

```
Name: &nameVault,
Billing: billingVault,
Resources: listResourcesVault,
Description: &descriptionVault,
BackupPolicyId: &backupPolicyIdVault,
Tags: &listTagsVault,
AutoBind: &autoBindVault,
BindRules: bindRulesVault,
Threshold: &thresholdVault,
SmnNotify: &smnNotifyVault,
Parameters: parametersVault,
}
request.Body = &model.VaultOrderCreateReqs{
    Vault: vaultbody,
}
response, err := client.CreatePostPaidVault(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.5.15 存储库容量总览

功能介绍

查询项目下所有存储库的总容量和总使用量

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vaults/summary

表 4-273 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

无

响应参数

状态码： 200

表 4-274 响应 Body 参数

参数	参数类型	描述
size	Integer	总容量大小 最小值： 0
used_size	Integer	总使用量 最小值： 0

请求示例

```
GET https://{{endpoint}}/v3/{{{project_id}}}/vaults/summary
```

响应示例

状态码： 200

OK

```
{  
    "size" : 27670,  
    "used_size" : 43  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;
```

```
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowSummarySolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowSummaryRequest request = new ShowSummaryRequest();
        try {
            ShowSummaryResponse response = client.showSummary(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \\\n
        client = CbrClient.new_builder() \\
            .with_credentials(credentials) \\
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \\
            .build()

    try:
        request = ShowSummaryRequest()
        response = client.show_summary(request)
        print(response)
```

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

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
    request := &model.ShowSummaryRequest{}  
    response, err := client.ShowSummary(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6 文件应用备份

4.6.1 新增备份路径

功能介绍

对客户端新增备份路径，新增的路径不会校验是否存在。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/agents/{agent_id}/add-path

表 4-275 路径参数

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
project_id	是	String	项目ID

请求参数

表 4-276 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-277 请求 Body 参数

参数	是否必选	参数类型	描述
add_path	是	Array of strings	增加备份路径详情

响应参数

状态码： 200

表 4-278 响应 Body 参数

参数	参数类型	描述
added	Array of strings	新添加成功的路径列表
existed	Array of strings	已经存在的路径列表

请求示例

```
POST https://{{endpoint}}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/agents/79bd9daa-884f-4f84-b8fe-235d58cd927d/add-path

{
  "add_path" : [ "/home/hcp", "/opt/workspace/" ]
```

响应示例

状态码： 200

OK

```
{
  "added" : [ "/opt/backup", "/var/log" ],
  "existed" : [ "/tmp/backup" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class AddAgentPathSolution {

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

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

AddAgentPathRequest request = new AddAgentPathRequest();
AgentAddPathReq body = new AgentAddPathReq();
List<String> listbodyAddPath = new ArrayList<>();
listbodyAddPath.add("/home/hcp");
listbodyAddPath.add("/opt/workspace/");
body.withAddPath(listbodyAddPath);
request.withBody(body);
try {
    AddAgentPathResponse response = client.addAgentPath(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = AddAgentPathRequest()
        listAddPathbody = [
            "/home/hcp",
            "/opt/workspace/"
        ]
        request.body = AgentAddPathReq(
            add_path=listAddPathbody
        )
        response = client.add_agent_path(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddAgentPathRequest{}
    var listAddPathbody = []string{
        "/home/hcp",
        "/opt/workspace/",
    }
    request.Body = &model.AgentAddPathReq{
        AddPath: listAddPathbody,
    }
    response, err := client.AddAgentPath(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.2 查询指定客户端

功能介绍

查询指定客户端

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/agents/{agent_id}

表 4-279 路径参数

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
project_id	是	String	项目ID

请求参数

表 4-280 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-281 响应 Body 参数

参数	参数类型	描述
agent	Agent object	客户端详情

表 4-282 Agent

参数	参数类型	描述
created_at	String	客户端创建时间
updated_at	String	客户端更新时间
agent_id	String	客户端ID
agent_version	String	客户端版本号
agent_type	String	客户端类型
host_name	String	客户端所在的主机名
host_nickname	String	客户端所在的主机昵称
host_ip	String	客户端所在主机的IP
host_os	String	客户端主机所在的操作系统
status	String	客户端状态
last_active_time	String	客户端上次激活时间
paths	Array of Path objects	客户端的备份路径

表 4-283 Path

参数	参数类型	描述
id	String	路径ID
status	String	路径状态，有available和remove两种状态
agent_id	String	该路径所属于的客户端ID
dir_path	String	路径详情

请求示例

```
GET https://{endpoint}/v3/{project_id}/agents/{agent_id}
```

响应示例

状态码： 200

OK

```
{
  "agent": {
    "created_at": "2022-01-30T09:28:09.334+00:00",
    "updated_at": "2022-01-30T09:28:09.334+00:00",
    "agent_id": "agent-12345678901234567890123456789012",
    "agent_version": "1.0.0",
    "agent_type": "Windows",
    "host_name": "Host-001",
    "host_nickname": "My-Host",
    "host_ip": "192.168.1.100",
    "host_os": "Windows 10 Pro",
    "status": "active",
    "last_active_time": "2022-01-30T09:28:09.334+00:00"
  }
}
```

```
"agent_id" : "489654f2-363b-4ebf-966a-bdf079d97114",
"agent_version" : "1.0.0",
"agent_type" : "native",
"host_name" : "ecs-0001",
"hostNickname" : "",
"host_ip" : "192.168.10.144",
"host_os" : "linux",
"status" : "normal",
"last_active_time" : "2022-01-30T09:28:08.960+00:00",
"paths" : [ {
    "id" : "5684b0fb-44c8-4c70-bb59-b03ec2738360",
    "agent_id" : "0cba3ff9-f836-4178-9ce8-91dc8026321c",
    "dir_path" : "/opt/huaweicloud/cbragent/lib",
    "status" : "available"
}, {
    "id" : "1ad949d5-c955-4995-8c28-60fc8b110a21",
    "agent_id" : "0cba3ff9-f836-4178-9ce8-91dc8026321c",
    "dir_path" : "/opt/backup",
    "status" : "available"
} ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowAgentSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowAgentRequest request = new ShowAgentRequest();
        try {
            ShowAgentResponse response = client.showAgent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

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

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.3 修改客户端

功能介绍

修改客户端状态

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/agents/{agent_id}

表 4-284 路径参数

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
project_id	是	String	项目ID

请求参数

表 4-285 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-286 请求 Body 参数

参数	是否必选	参数类型	描述
agent	是	AgentUpdate object	待更新的agent参数

表 4-287 AgentUpdate

参数	是否必选	参数类型	描述
status	是	String	客户端状态，当前只支持卸载，由客户端被卸载时自动触发 枚举值： <ul style="list-style-type: none">• uninstall

响应参数

状态码： 200

表 4-288 响应 Body 参数

参数	参数类型	描述
agent	Agent object	客户端详情

表 4-289 Agent

参数	参数类型	描述
created_at	String	客户端创建时间
updated_at	String	客户端更新时间
agent_id	String	客户端ID

参数	参数类型	描述
agent_version	String	客户端版本号
agent_type	String	客户端类型
host_name	String	客户端所在的主机名
host_nickname	String	客户端所在的主机昵称
host_ip	String	客户端所在主机的IP
host_os	String	客户端主机所在的操作系统
status	String	客户端状态
last_active_time	String	客户端上次激活时间
paths	Array of Path objects	客户端的备份路径

表 4-290 Path

参数	参数类型	描述
id	String	路径ID
status	String	路径状态，有available和remove两种状态
agent_id	String	该路径所属于的客户端ID
dir_path	String	路径详情

请求示例

更新客户端状态为卸载

```
PUT https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/agents/79bd9daa-884f-4f84-b8fe-235d58cd927d
{
  "agent": {
    "status": "uninstall"
  }
}
```

响应示例

状态码： 200

OK

```
{
  "agent": {
    "created_at": "2022-01-30T09:28:09.334+00:00",
    "version": "1.0.0"
  }
}
```

```
"updated_at": "2022-01-30T09:28:09.334+00:00",
"agent_id": "489654f2-363b-4ebf-966a-bdf079d97114",
"agent_version": "1.0.0",
"agent_type": "native",
"host_name": "ecs-0001",
"hostNickname": "",
"host_ip": "192.168.10.144",
"host_os": "linux",
"status": "uninstall",
"last_active_time": "2022-01-30T09:28:08.960+00:00",
"paths": [ {
    "id": "5684b0fb-44c8-4c70-bb59-b03ec2738360",
    "agent_id": "0cba3ff9-f836-4178-9ce8-91dc8026321c",
    "dir_path": "/opt/huaweicloud/cbragent/lib",
    "status": "available"
}, {
    "id": "1ad949d5-c955-4995-8c28-60fc8b110a21",
    "agent_id": "0cba3ff9-f836-4178-9ce8-91dc8026321c",
    "dir_path": "/opt/backup",
    "status": "available"
} ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新客户端状态为卸载

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class UpdateAgentSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAgentRequest request = new UpdateAgentRequest();
        AgentUpdateReq body = new AgentUpdateReq();
        AgentUpdate agentbody = new AgentUpdate();
        agentbody.withStatus(AgentUpdate.StatusEnum.fromValue("uninstall"));
    }
}
```

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

Python

更新客户端状态为卸载

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateAgentRequest()
        agentbody = AgentUpdate(
            status="uninstall"
        )
        request.body = AgentUpdateReq(
            agent=agentbody
        )
        response = client.update_agent(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新客户端状态为卸载

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
}

request := &model.UpdateAgentRequest{}
agentbody := &model.AgentUpdate{
    Status: model.GetAgentUpdateStatusEnum().UNINSTALL,
}
request.Body = &model.AgentUpdateReq{
    Agent: agentbody,
}
response, err := client.UpdateAgent(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.4 注册客户端

功能介绍

注册客户端，安装时候由Agent调用，无需手动注册。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/agents

表 4-291 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-292 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用。

表 4-293 请求 Body 参数

参数	是否必选	参数类型	描述
agent	是	AgentRegister object	客户端注册参数

表 4-294 AgentRegister

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
host_name	是	String	客户端所在的主机名
host_ip	是	String	客户端所在主机的IP
host_os	是	String	客户端所在主机的操作系统

参数	是否必选	参数类型	描述
host_nickname	否	String	客户端所在主机的主机别名
agent_version	否	String	客户端版本
agent_type	否	String	客户端类型，分本地客户端和云上客户端(cloud/native)

响应参数

状态码： 200

表 4-295 响应 Body 参数

参数	参数类型	描述
agent	Agent object	客户端详情

表 4-296 Agent

参数	参数类型	描述
created_at	String	客户端创建时间
updated_at	String	客户端更新时间
agent_id	String	客户端ID
agent_version	String	客户端版本号
agent_type	String	客户端类型
host_name	String	客户端所在的主机名
host_nickname	String	客户端所在的主机昵称
host_ip	String	客户端所在主机的IP
host_os	String	客户端主机所在的操作系统
status	String	客户端状态
last_active_time	String	客户端上次激活时间
paths	Array of Path objects	客户端的备份路径

表 4-297 Path

参数	参数类型	描述
id	String	路径ID
status	String	路径状态，有available和remove两种状态
agent_id	String	该路径所属于的客户端ID
dir_path	String	路径详情

请求示例

注册客户端由agent安装时自动触发，无需客户手动调用。

```
POST https://{{endpoint}}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/agents
```

```
{  
    "agent": {  
        "agent_id": "489654f2-363b-4ebf-966a-bdf079d97114",  
        "host_name": "ecs-00002",  
        "host_ip": "10.10.0.2",  
        "host_os": "linux",  
        "agent_version": "1.0.0",  
        "agent_type": "native"  
    }  
}
```

响应示例

状态码： 200

OK

```
{  
    "agent": {  
        "created_at": "2022-01-30T09:28:09.334+00:00",  
        "updated_at": "2022-01-30T09:28:09.334+00:00",  
        "agent_id": "489654f2-363b-4ebf-966a-bdf079d97114",  
        "agent_version": "1.0.0",  
        "agent_type": "native",  
        "host_name": "ecs-0001",  
        "host_nickname": "",  
        "host_ip": "192.168.10.144",  
        "host_os": "linux",  
        "status": "normal",  
        "last_active_time": "2022-01-30T09:28:08.960+00:00"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

注册客户端由agent安装时自动触发，无需客户手动调用。

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class RegisterAgentSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        RegisterAgentRequest request = new RegisterAgentRequest();
        AgentRegisterReq body = new AgentRegisterReq();
        AgentRegister agentbody = new AgentRegister();
        agentbody.withAgentId("489654f2-363b-4ebf-966a-bdf079d97114")
            .withHostName("ecs-00002")
            .withHostIp("10.10.0.2")
            .withHostOs("linux")
            .withAgentVersion("1.0.0")
            .withAgentType("native");
        body.withAgent(agentbody);
        request.withBody(body);
        try {
            RegisterAgentResponse response = client.registerAgent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

注册客户端由agent安装时自动触发，无需客户手动调用。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \


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

try:
    request = RegisterAgentRequest()
    agentbody = AgentRegister(
        agent_id="489654f2-363b-4ebf-966a-bdf079d97114",
        host_name="ecs-00002",
        host_ip="10.10.0.2",
        host_os="linux",
        agent_version="1.0.0",
        agent_type="native"
    )
    request.body = AgentRegisterReq(
        agent=agentbody
    )
    response = client.register_agent(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

注册客户端由agent安装时自动触发，无需客户手动调用。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region" "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
}
```

```
request := &model.RegisterAgentRequest{}
agentVersionAgent:= "1.0.0"
agentTypeAgent:= "native"
agentbody := &model.AgentRegister{
    AgentId: "489654f2-363b-4ebf-966a-bdf079d97114",
    HostName: "ecs-00002",
    HostIp: "10.10.0.2",
    HostOs: "linux",
    AgentVersion: &agentVersionAgent,
    AgentType: &agentTypeAgent,
}
request.Body = &model.AgentRegisterReq{
    Agent: agentbody,
}
response, err := client.RegisterAgent(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.5 移除客户端

功能介绍

移除客户端，移除客户端时将会删除该客户端所有备份，请谨慎操作。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/agents/{agent_id}

表 4-298 路径参数

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
project_id	是	String	项目ID

请求参数

表 4-299 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

无

请求示例

解除注册客户端

```
DELETE https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/agents/6dd81d7d-a4cb-443e-b8ed-1af0bd3a261b
```

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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.  
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_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);  
  
CbrClient client = CbrClient.newBuilder()  
    .withCredential(auth)  
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
    .build();  
UnregisterAgentRequest request = new UnregisterAgentRequest();  
try {  
    UnregisterAgentResponse response = client.unregisterAgent(request);  
    System.out.println(response.toString());  
} catch (ConnectionException e) {  
    e.printStackTrace();  
} catch (RequestTimeoutException e) {  
    e.printStackTrace();  
} catch (ServiceResponseException e) {  
    e.printStackTrace();  
    System.out.println(e.getHttpStatusCode());  
    System.out.println(e.getRequestId());  
    System.out.println(e.getErrorCode());  
    System.out.println(e.getErrorMsg());  
}  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CbrClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CbrRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UnregisterAgentRequest()  
        response = client.unregister_agent(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.6 移除备份路径

功能介绍

移除已添加的文件备份路径。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/agents/{agent_id}/remove-path

表 4-300 路径参数

参数	是否必选	参数类型	描述
agent_id	是	String	客户端ID
project_id	是	String	项目ID

请求参数

表 4-301 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-302 请求 Body 参数

参数	是否必选	参数类型	描述
remove_path	是	Array of strings	移除备份路径详情

响应参数

状态码： 200

表 4-303 响应 Body 参数

参数	参数类型	描述
removed	Array of strings	移除的路径列表
not_existed	Array of strings	不存在的路径

请求示例

移除备份路径

```
https://{{endpoint}}/v3/{{project_id}}/agents/{{agent_id}}/remove-path
{
  "remove_path" : [ "/tmp", "/home" ]
```

响应示例

状态码： 200

OK

```
{
  "removed" : [ "/tmp" ],
  "not_existed" : [ "/home" ]}
```

SDK 代码示例

SDK代码示例如下。

Java

移除备份路径

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class RemoveAgentPathSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        RemoveAgentPathRequest request = new RemoveAgentPathRequest();
        AgentRemovePathReq body = new AgentRemovePathReq();
        List<String> listbodyRemovePath = new ArrayList<>();
        listbodyRemovePath.add("/tmp");
```

```
listbodyRemovePath.add("/home");
body.withRemovePath(listbodyRemovePath);
request.withBody(body);
try {
    RemoveAgentPathResponse response = client.removeAgentPath(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

移除备份路径

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = RemoveAgentPathRequest()
        listRemovePathbody = [
            "/tmp",
            "/home"
        ]
        request.body = AgentRemovePathReq(
            remove_path=listRemovePathbody
        )
        response = client.remove_agent_path(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

移除备份路径

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.RemoveAgentPathRequest{}
    var listRemovePathbody = []string{
        "/tmp",
        "/home",
    }
    request.Body = &model.AgentRemovePathReq{
        RemovePath: listRemovePathbody,
    }
    response, err := client.RemoveAgentPath(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.6.7 查询客户端列表

功能介绍

查询客户端列表

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/agents

表 4-304 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-305 Query 参数

参数	是否必选	参数类型	描述
agent_id	否	String	客户端ID
limit	否	String	每页显示条目数，正整数 最小长度：1 最大长度：1000
offset	否	Integer	偏移值,正整数
status	否	String	状态

请求参数

表 4-306 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-307 响应 Body 参数

参数	参数类型	描述
agents	Array of Agent objects	客户端实例列表
count	Integer	客户端个数
limit	Integer	每页显示的条目数量
offset	Integer	偏移量，表示从此偏移量开始查询

表 4-308 Agent

参数	参数类型	描述
created_at	String	客户端创建时间
updated_at	String	客户端更新时间
agent_id	String	客户端ID
agent_version	String	客户端版本号
agent_type	String	客户端类型
host_name	String	客户端所在的主机名
host_nickname	String	客户端所在的主机昵称
host_ip	String	客户端所在主机的IP
host_os	String	客户端主机所在的操作系统
status	String	客户端状态
last_active_time	String	客户端上次激活时间
paths	Array of Path objects	客户端的备份路径

表 4-309 Path

参数	参数类型	描述
id	String	路径ID
status	String	路径状态，有available和remove两种状态
agent_id	String	该路径所属于的客户端ID
dir_path	String	路径详情

请求示例

```
GET https://{{endpoint}}/v3/{{project_id}}/agents
```

响应示例

状态码： 200

OK

```
{  
    "count" : 2,  
    "limit" : 1000,  
    "offset" : 0,  
    "agents" : [ {  
        "created_at" : "2022-01-30T09:28:09.334+00:00",  
        "agent_id" : "489654f2-363b-4ebf-966a-bdf079d97114",  
        "host_name" : "eBackup",  
        "host_ip" : "18.9.158.36",  
        "status" : "normal",  
        "last_active_time" : "2022-01-30T09:28:08.960+00:00"  
    }, {  
        "created_at" : "2022-01-14T06:47:17.524+00:00",  
        "updated_at" : "2022-01-28T05:01:32.739+00:00",  
        "agent_id" : "fb711278-e872-4224-b442-8dafc36a543d",  
        "host_name" : "ecs-00001",  
        "host_ip" : "10.10.0.1",  
        "status" : "normal",  
        "last_active_time" : "2022-01-28T05:01:32.000+00:00",  
        "paths" : [ {  
            "id" : "6c3d1ed6-b7d7-4e27-94ee-321431e84c5c",  
            "dir_path" : "/home/hcp",  
            "status" : "available",  
            "agent_id" : "fb711278-e872-4224-b442-8dafc36a543d"  
        }, {  
            "id" : "e2af0fdd-9644-4db8-986e-5a6a2395c390",  
            "dir_path" : "/opt/workspace",  
            "status" : "available",  
            "agent_id" : "fb711278-e872-4224-b442-8dafc36a543d"  
        } ]  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListAgentSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

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

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_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);

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

ListAgentRequest request = new ListAgentRequest();
request.withLimit("<limit>");
request.withOffset(<offset>);
request.withStatus("<status>");
request.withAgentId();
try {
    ListAgentResponse response = client.listAgent(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \
        client = CbrClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \
            .build()

    try:
        request = ListAgentRequest()
        request.limit = "<limit>"
        request.offset = <offset>
        request.status = "<status>"
        request.agent_id =
        response = client.list_agent(request)
```

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build())

    request := &model.ListAgentRequest{}
    limitRequest:= "<limit>"
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    statusRequest:= "<status>"
    request.Status = &statusRequest
    response, err := client.ListAgent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.7 标签

4.7.1 查询存储库资源实例

功能介绍

使用标签过滤实例 标签管理服务需要提供按标签过滤各服务实例并汇总显示在列表中，需要各服务提供查询能力

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vault/resource_instances/action

表 4-310 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id

请求参数

表 4-311 请求 Body 参数

参数	是否必选	参数类型	描述
without_any_tag	否	Boolean	不包含任意一个标签，该字段为true时查询所有不带标签的资源，此时忽略“tags”、“tags_any”、“not_tags”、“not_tags_any”字段。

参数	是否必选	参数类型	描述
tags	否	Array of TagsReq objects	包含标签。 tags不允许为空列表。 tags中最多包含10个key。 tags中key不允许重复。 tags中多个key之间是“与”的关系。 结果返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。 无过滤条件时返回全量数据。
tags_any	否	Array of TagsReq objects	包含任一标签。 tags不允许为空列表。 tags中最多包含10个key。 tags中key不允许重复。 结果返回包含任一标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。 无过滤条件时返回全量数据。
not_tags	否	Array of TagsReq objects	不包含标签。 tags不允许为空列表。 tags中最多包含10个key。 tags中key不允许重复。 结果返回不包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。 无过滤条件时返回全量数据。
not_tags_any	否	Array of TagsReq objects	不包含任一标签。 tags不允许为空列表。 tags中最多包含10个key。 tags中key不允许重复。 结果返回不包含任一标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。 无过滤条件时返回全量数据。

参数	是否必选	参数类型	描述
sys_tags	否	Array of SysTags objects	<p>仅op_service权限可以使用此字段做资源实例过滤条件。</p> <p>目前TMS调用时只包含一个tag结构体。</p> <ul style="list-style-type: none">• key: _sys_enterprise_project_id• values: 企业项目id列表 <p>目前TMS调用时, key下面只包含一个value, 0表示默认企业项目。</p> <p>sys_tags和租户标签过滤条件 (tags、tags_any、not_tags、not_tags_any) 不能同时使用。</p> <p>无sys_tags时按照tag接口处理, 无tag过滤条件时返回全量数据。</p> <p>sys_tags不能为空列表</p>
limit	否	String	查询记录数 (action为count时无此参数) 如果action为filter时, 默认为1000, limit最小值为1, limit最大值为1000, 不在范围内报错。返回的结果中记录数不超过limit。
offset	否	String	索引位置 (action为count时无此参数) 如果action为filter时, 默认为0, offset最小值为0。返回的结果中第一条记录为符合查询条件的第offset+1条记录。
action	是	String	操作标识取值范围为: "filter", "count"。如果是filter就是分页查询, 如果是count只需按照条件将总条数返回即可
matches	否	Array of Match objects	资源本身支持的查询条件。 matches中key不允许重复。 数组长度最大值为1, 后续再扩展。 数组长度: 0 - 1
cloud_type	否	String	云类型 枚举值: <ul style="list-style-type: none">• public• hybrid

参数	是否必选	参数类型	描述
object_type	否	String	资源类型 枚举值： <ul style="list-style-type: none">serverdisk

表 4-312 TagsReq

参数	是否必选	参数类型	描述
key	是	String	键。 最大长度127个unicode字符。 不允许为空字符串。 前后的空格会被丢弃。
values	是	Array of strings	值列表。 values中最多包含10个value。 每个value最大长度255个 unicode字符。前后的空格会被 丢弃。 values中value不允许重复。 values中多个value之间是"或" 的关系。 values允许为空列表， value允 许为空字符串。 values如果为空列表， 表示任意 值。 *为系统保留字符， 如果value是 以*开头表示按照*后面的值全模 糊匹配， 不能只传入“*”。

表 4-313 SysTags

参数	是否必选	参数类型	描述
key	是	String	键。 系统标签的key， 从白名单 中取， 不能随意定义。 目前仅 支持 _sys_enterprise_project_id 字段， 对应 的value为企业项目 ID。
values	是	Array of strings	值列表。 目前仅会用到企业项 目ID， 其中默 认的企业项目ID 为“0”。

表 4-314 Match

参数	是否必选	参数类型	描述
key	是	String	键。key取值范围为："resource_name"，资源名称
value	是	String	值。最大长度255个字符。key为"resource_name"时，value为空字符串时精确匹配，为非空字符串时模糊匹配。

响应参数

状态码： 200

表 4-315 响应 Body 参数

参数	参数类型	描述
resources	Array of TagResource objects	符合查询条件的资源列表（action为count时无此参数）。
total_count	Integer	符合查询条件的资源总个数

表 4-316 TagResource

参数	参数类型	描述
resource_id	String	资源ID
resource_detail	InstancesResourceDetail object	资源详情
tags	Array of Tag objects	标签列表 没有标签默认为空数组。
resource_name	String	资源名称
sys_tags	Array of SysTag objects	仅op_service权限才可以获取此字段： 目前只包含一个resource_tag 结构体。 key: _sys_enterprise_project_id value: 企业项目id, 0表示默认企业项目 非op_service场景不能返回此字段。

表 4-317 InstancesResourceDetail

参数	参数类型	描述
vault	Vault object	存储库

表 4-318 Vault

参数	参数类型	描述
billing	Billing object	运营信息
description	String	存储库自定义描述信息。 最小长度: 0 最大长度: 255
id	String	存储库ID
name	String	存储库名称 最小长度: 1 最大长度: 64
project_id	String	项目ID
provider_id	String	存储库资源类型id
resources	Array of ResourceRes p objects	存储库资源
tags	Array of Tag objects	存储库标签
enterprise_project_id	String	企业项目id， 默认为 ‘0’。
auto_bind	Boolean	是否自动绑定， 默认为false， 不支持。
bind_rules	VaultBindRules object	绑定规则
user_id	String	用户id
created_at	String	创建时间,例如:"2020-02-05T10:38:34.209782"
auto_expand	Boolean	是否开启存储库自动扩容能力（只支持按需存储库）。
smn_notify	Boolean	存储库smn消息通知开关 缺省值: true

参数	参数类型	描述
threshold	Integer	存储库容量阈值，已用容量占总容量达到此百分比即发送相关通知 最小值：1 最大值：100 缺省值：80
sys_lock_source_service	String	用于标识SMB服务 最小长度：0 最大长度：32 枚举值： <ul style="list-style-type: none">• SMB• "

表 4-319 Billing

参数	参数类型	描述
allocated	Integer	已分配容量，单位GB
charging_mode	String	创建模式，按需：post_paid，包周期：pre_paid，默认为post_paid
cloud_type	String	公有云:public ;混合云:hybrid
consistent_level	String	规格，崩溃一致性 (crash_consistent) 或应用一致性 (app_consistent)
object_type	String	对象类型：云服务器 (server)，云硬盘 (disk)，文件系统 (turbo)，云桌面 (workspace)，VMware (vmware)，关系型数据库 (rds)，文件 (file)。
order_id	String	订单ID
product_id	String	产品ID
protect_type	String	保护类型：备份 (backup)、复制 (replication)。
size	Integer	容量，单位GB 最小值：1 最大值：10485760
spec_code	String	规格编码。云服务备份存储库：vault.backup.server.normal；云硬盘备份存储库：vault.backup.volume.normal；文件备份存储库：vault.backup.turbo.normal

参数	参数类型	描述
status	String	存储库状态 枚举值： <ul style="list-style-type: none">• available• lock• frozen• deleting• error
storage_unit	String	存储库桶名
used	Integer	已使用容量, 单位MB
frozen_scene	String	冻结场景
is_multi_az	Boolean	存储库多az属性 缺省值: false

表 4-320 ResourceResp

参数	参数类型	描述
extra_info	ResourceExtraInfo object	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称 最小长度: 0 最大长度: 255
protect_status	String	保护状态 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
size	Integer	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	Integer	副本大小

参数	参数类型	描述
backup_count	Integer	副本数量

表 4-321 ResourceExtraInfo

参数	参数类型	描述
exclude_volumes	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有 效，排除不需要备份的磁盘。当虚拟机新绑定磁 盘时，也能继续排除之前设置不用备份的卷。

表 4-322 VaultBindRules

参数	参数类型	描述
tags	Array of BindRulesTags objects	按tags过滤自动绑定的资源 最小长度：0 最大长度：5 数组长度：0 - 5

表 4-323 BindRulesTags

参数	参数类型	描述
key	String	key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-324 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-325 SysTag

参数	参数类型	描述
key	String	键。系统标签的key, 从白名单中取, 不能随意定义。目前仅支持 _sys_enterprise_project_id字段, 对应的value为企业项目ID。
value	String	值。目前仅会用到企业项目ID, 其中默认的企业项目ID为“0”。

状态码: 400

表 4-326 响应 Body 参数

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

请求示例

使用标签分页查询存储库资源实例。

```
POST https://{{endpoint}}/v3/{{project_id}}/vault/resource_instances/action
{
  "tags": [ {
    "key": "string",
    "values": [ "value" ]
  }],
  "action": "filter"
}
```

响应示例

状态码： 200

OK

```
{
  "total_count": 7,
  "resources": [ {
    "resource_id": "e54f7854-8de8-4f98-acf1-65f330bfe877",
    "resource_name": "vault-7698-api-test",
    "resource_detail": {
      "vault": {
        "id": "e54f7854-8de8-4f98-acf1-65f330bfe877",
        "name": "vault-7698-api-test",
        "resources": [ ],
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",
        "created_at": "2023-04-21T08:20:43.022+00:00",
        "project_id": "667230ba418f48f1996329f174693053",
        "enterprise_project_id": 0,
        "auto_bind": false,
        "bind_rules": { },
        "auto_expand": false,
        "smn_notify": true,
        "threshold": 80,
        "user_id": "0ae13f012b80d2d81f11c0129285d34b",
        "billing": {
          "allocated": 0,
          "cloud_type": "public",
          "consistent_level": "crash_consistent",
          "charging_mode": "pre_paid",
          "order_id": "CS23042116192E6CZ",
          "product_id": "00301-231151-0--0",
          "protect_type": "backup",
          "object_type": "server",
          "spec_code": "vault.backup.server.normal",
          "used": 0,
          "status": "available",
          "size": 100
        },
        "tags": [ ]
      }
    },
    "tags": [ ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

使用标签分页查询存储库资源实例。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class ShowVaultResourceInstancesSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowVaultResourceInstancesRequest request = new ShowVaultResourceInstancesRequest();
        VaultResourceInstancesReq body = new VaultResourceInstancesReq();
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("vaule");
        List<TagsReq> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagsReq()
                .withKey("string")
                .withValues(listTagsValues)
        );
        body.withAction("filter");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            ShowVaultResourceInstancesResponse response = client.showVaultResourceInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

使用标签分页查询存储库资源实例。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowVaultResourceInstancesRequest()
        listValuesTags = [
            "vaule"
        ]
        listTagsbody = [
            TagsReq(
                key="string",
                values=listValuesTags
            )
        ]
        request.body = VaultResourceInstancesReq(
            action="filter",
            tags=listTagsbody
        )
        response = client.show_vault_resource_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

使用标签分页查询存储库资源实例。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>").
        WithCredential(auth).
        Build())

request := &model.ShowVaultResourceInstancesRequest{}
var listValuesTags = []string{
    "vaule",
}
var listTagsbody = []model.TagsReq{
{
    Key: "string",
    Values: listValuesTags,
},
}
request.Body = &model.VaultResourceInstancesReq{
    Action: "filter",
    Tags: &listTagsbody,
}
response, err := client.ShowVaultResourceInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.7.2 批量添加删除存储库资源标签

功能介绍

为指定实例批量添加或删除标签 标签管理服务需要使用该接口批量管理实例的标签。一个资源上最多有10个标签。此接口为幂等接口：创建时如果请求体中存在重复key

则报错。创建时，不允许重复key，如果数据库存在就覆盖。删除时，允许重复key。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127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。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vault/{vault_id}/tags/action

表 4-327 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
vault_id	是	String	资源id

请求参数

表 4-328 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of Tag objects	标签列表。 tags不允许为空列表。 tags中最多包含10个key。 tags中key不允许重复。
sys_tags	否	Array of SysTag objects	系统标签列表。 op_service权限可以访问，和tags二选一。 目前TMS调用时只包含一个resource_tag结构体，key固定为： _sys_enterprise_project_id。 value是UUID或0,value为0表示默认企业项目。 现在仅支持create操作。
action	是	String	操作标识：仅限于create（创建）、delete（删除） 枚举值： • create • delete

表 4-329 Tag

参数	是否必选	参数类型	描述
key	是	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	是	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符 ASCII(0-31), “=” “*” “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-330 SysTag

参数	是否必选	参数类型	描述
key	是	String	键。系统标签的key, 从白名单中取, 不能随意定义。目前仅支持 _sys_enterprise_project_id 字段, 对应 的value为企业项目ID。
value	是	String	值。目前仅会用到企业项目ID, 其中默认的企业项目ID为“0”。

响应参数

状态码: 400

表 4-331 响应 Body 参数

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

请求示例

批量创建tag

```
POST https://[endpoint]/v3/{project_id}/vault/{vault_id}/tags/action
{
  "tags": [
    {
      "key": "string",
      "value": "string"
    },
    {
      "key": "string1",
      "value": "string2"
    }
  ],
  "action": "create"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

批量创建tag

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class BatchCreateAndDeleteVaultTagsSolution {

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

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
BatchCreateAndDeleteVaultTagsRequest request = new BatchCreateAndDeleteVaultTagsRequest();
BulkCreateAndDeleteVaultTagsReq body = new BulkCreateAndDeleteVaultTagsReq();
List<Tag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tag()
        .withKey("string")
        .withValue("string")
);
listbodyTags.add(
    new Tag()
        .withKey("string1")
        .withValue("string2")
);
body.withAction(BulkCreateAndDeleteVaultTagsReq.ActionEnum.fromValue("create"));
body.withTags(listbodyTags);
request.withBody(body);
try {
    BatchCreateAndDeleteVaultTagsResponse response =
client.batchCreateAndDeleteVaultTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

批量创建tag

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \\\
        client = CbrClient.new_builder() \\
            .with_credentials(credentials) \\
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \\
            .build()
```

```
try:
    request = BatchCreateAndDeleteVaultTagsRequest()
    listTagsbody = [
        Tag(
            key="string",
            value="string"
        ),
        Tag(
            key="string1",
            value="string2"
        )
    ]
    request.body = BulkCreateAndDeleteVaultTagsReq(
        action="create",
        tags=listTagsbody
    )
    response = client.batch_create_and_delete_vault_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

批量创建tag

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateAndDeleteVaultTagsRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "string",
            Value: "string",
        },
        {
            Key: "string1",
            Value: "string2",
        },
    }
```

```
        }
        request.Body = &model.BulkCreateAndDeleteVaultTagsReq{
            Action: model.GetBulkCreateAndDeleteVaultTagsReqActionEnum().CREATE,
            Tags: &listTagsbody,
        }
        response, err := client.BatchCreateAndDeleteVaultTags(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
```

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request

错误码

请参见[错误码](#)。

4.7.3 添加存储库资源标签

功能介绍

一个资源上最多有10个标签。此接口为幂等接口：创建时，如果创建的标签已经存在（key相同），则覆盖。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/vault/{vault_id}/tags

表 4-332 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
vault_id	是	String	资源id

请求参数

表 4-333 请求 Body 参数

参数	是否必选	参数类型	描述
tag	否	Tag object	标签

表 4-334 Tag

参数	是否必选	参数类型	描述
key	是	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文，字母，数字， “-” , “_” 组成。
value	是	String	值。 添加标签时value值必选，删除 标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符 ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文，字母，数 字， “-” , “_” , “.” 组成。

响应参数

状态码： 400

表 4-335 响应 Body 参数

参数	参数类型	描述
error_code	String	请参见 错误码 。

参数	参数类型	描述
error_msg	String	错误信息

请求示例

添加存储库标签

```
POST https://{{endpoint}}/v3/{{project_id}}/vault/{{vault_id}}/tags
{
  "tag": {
    "key": "key1",
    "value": "key2"
  }
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

添加存储库标签

```
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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class CreateVaultTagsSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateVaultTagsRequest request = new CreateVaultTagsRequest();
```

```
VaultTagsCreateReq body = new VaultTagsCreateReq();
Tag tagbody = new Tag();
tagbody.withKey("key1")
    .withValue("key2");
body.withTag(tagbody);
request.withBody(body);
try {
    CreateVaultTagsResponse response = client.createVaultTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

添加存储库标签

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateVaultTagsRequest()
        tagbody = Tag(
            key="key1",
            value="key2"
        )
        request.body = VaultTagsCreateReq(
            tag=tagbody
        )
        response = client.create_vault_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

添加存储库标签

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.CreateVaultTagsRequest{}
    tagbody := &model.Tag{
        Key: "key1",
        Value: "key2",
    }
    request.Body = &model.VaultTagsCreateReq{
        Tag: tagbody,
    }
    response, err := client.CreateVaultTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request

错误码

请参见[错误码](#)。

4.7.4 删除存储库资源标签

功能介绍

幂等接口：删除时，如果删除的标签不存在，返回404。Key不能为空或者空字符串。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/vault/{vault_id}/tags/{key}

表 4-336 路径参数

参数	是否必选	参数类型	描述
key	是	String	不能为空或空字符串，不检查长度和字符集，去掉key前后的空格后检查，去掉key前后的空格后使用。即使底层存在非法的tag也要能删。
project_id	是	String	项目id
vault_id	是	String	资源id

请求参数

无

响应参数

状态码： 400

表 4-337 响应 Body 参数

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

请求示例

```
DELETE https://{{endpoint}}/v3/{{project_id}}/vault/{{vault_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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DeleteVaultTagSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteVaultTagRequest request = new DeleteVaultTagRequest();
        try {
            DeleteVaultTagResponse response = client.deleteVaultTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteVaultTagRequest()
        response = client.delete_vault_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"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
204	No Content
400	Bad Request

错误码

请参见[错误码](#)。

4.7.5 查询存储库资源标签

功能介绍

查询指定实例的标签信息 标签管理服务需要使用该接口查询指定实例的全部标签数据

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vault/{vault_id}/tags

表 4-338 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
vault_id	是	String	资源id

请求参数

无

响应参数

状态码： 200

表 4-339 响应 Body 参数

参数	参数类型	描述
tags	Array of Tag objects	标签列表 tags中key不重复
sys_tags	Array of SysTag objects	仅op_service权限才可以获取此字段： 目前只包含一个resource_tag结构体 key: _sys_enterprise_project_id value: 企业项目id。0表示默认企业项目 非op_service场景不能返回此字段。

表 4-340 Tag

参数	参数类型	描述
key	String	键。 key最大长度为36个字符。 key不能为空字符串。 key前后空格会被丢弃。 key不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 key只能由中文, 字母, 数字, “-” , “_” 组成。
value	String	值。 添加标签时value值必选, 删除标签时value值可选。 value最大长度为43个字符。 value可以为空字符串。 value前后的空格会被丢弃。 value不能包含非打印字符ASCII(0-31), “=” , “*” , “<” , “>” , “\” , “,” , “ ” , “/” 。 value只能由中文, 字母, 数字, “-” , “_” , “.” 组成。

表 4-341 SysTag

参数	参数类型	描述
key	String	键。 系统标签的key, 从白名单中取, 不能随意定义。 目前仅支持 _sys_enterprise_project_id字段, 对应 的value为企业项目ID。
value	String	值。 目前仅会用到企业项目ID, 其中默认的企业项目ID为“0”。

状态码： 400

表 4-342 响应 Body 参数

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

请求示例

GET https://{endpoint}/v3/{project_id}/vault/{vault_id}/tags

响应示例

状态码： 200

OK

```
{  
  "tags" : [ {  
    "key" : "string",  
    "value" : ""  
  } ]  
}
```

状态码： 400

Bad Request

```
{  
  "error_code" : "BackupService.9900",  
  "error_msg" : "Invalid vault_id provided."  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowVaultTagSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
    }  
}
```

```
this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ShowVaultTagRequest request = new ShowVaultTagRequest();
try {
    ShowVaultTagResponse response = client.showVaultTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowVaultTagRequest()
        response = client.show_vault_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"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

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

更多

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

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.7.6 查询存储库项目标签

功能介绍

查询租户在指定Region和实例类型的所有标签集合 标签管理服务需要能够列出当前租户全部已使用的标签集合，为各服务Console打标签和过滤实例时提供标签联想功能

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/vault/tags

表 4-343 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

无

响应参数

状态码： 200

表 4-344 响应 Body 参数

参数	参数类型	描述
tags	Array of TagsResp objects	标签列表

表 4-345 TagsResp

参数	参数类型	描述
key	String	键。 key最大长度36个字符。 key不能为空字符串。 key只能由中文，字母，数字，“-”，“_”组成。
values	String	值列表。 value最大长度43个字符。 value可以为空字符串。 key只能由中文，字母，数字，“-”，“_”组成。

状态码： 400

表 4-346 响应 Body 参数

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

请求示例

GET https://{endpoint}/v3/{project_id}/vault/tags

响应示例

状态码： 200

OK

```
{  
    "tags": [ {  
        "values": [ "b" ],  
        "key": "a"  
    }, {  
        "values": [ "", "string" ],  
        "key": "string"  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowVaultProjectTagSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
    }  
}
```

```
CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ShowVaultProjectTagRequest request = new ShowVaultProjectTagRequest();
try {
    ShowVaultProjectTagResponse response = client.showVaultProjectTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowVaultProjectTagRequest()
        response = client.show_vault_project_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"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	OK
400	Bad Request

错误码

请参见[错误码](#)。

4.8 策略

4.8.1 创建策略

功能介绍

创建策略，策略分为备份策略和复制策略。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/policies

表 4-347 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-348 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-349 请求 Body 参数

参数	是否必选	参数类型	描述
policy	是	PolicyCreate object	创建策略body

表 4-350 PolicyCreate

参数	是否必选	参数类型	描述
enabled	否	Boolean	是否启用策略 缺省值: true
name	是	String	策略名称，长度限制：1- 64， 只能由中文、字母、数字、 “_”、“-”组成。 最小长度： 1 最大长度： 64
operation_definition	是	PolicyODCreate object	调度参数

参数	是否必选	参数类型	描述
operation_type	是	String	保护类型：备份（ backup ）、复制(replication)。
trigger	是	PolicyTrigger Req object	策略执行时间规则

表 4-351 PolicyODCreate

参数	是否必选	参数类型	描述
day_backups	否	Integer	保留日备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone 也必选。 最小值：0 最大值：100
destination_project_id	否	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。
destination_region	否	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	否	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。
max_backups	否	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	否	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone 也必选。 最小值：0 最大值：100

参数	是否必选	参数类型	描述
retention_duration_days	否	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	否	String	用户所在时区,格式形如UTC +08:00, 若选择年备, 月备, 周备, 日备中任一参数, 则该参数不能为空。
week_backups	否	Integer	保留周备个数, 该备份不受保留最大备份数限制。取值为0到100。若选择该参数, 则timezone也必选。
year_backups	否	Integer	保留年备个数, 该备份不受保留最大备份数限制。取值为0到100。若选择该参数, 则timezone也必选。 最小值：0 最大值：100
full_backup_interval	否	Integer	每间隔多少次执行一次全量备份, 当取值为 -1 时, 不执行全量备份 最小值：-1 最大值：100

表 4-352 PolicyTriggerReq

参数	是否必选	参数类型	描述
properties	是	PolicyTriggerPropertiesRequest object	调度器属性

表 4-353 PolicyTriggerPropertiesReq

参数	是否必选	参数类型	描述
pattern	是	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天(MO、TU、WE、TH、FR、SA、SU)，BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为：'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00'。例如：某个地区的北京时间为UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为：'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。

响应参数

状态码： 200

表 4-354 响应 Body 参数

参数	参数类型	描述
policy	Policy object	创建响应

表 4-355 Policy

参数	参数类型	描述
enabled	Boolean	策略是否启用
id	String	策略ID
name	String	策略名称
operation_definition	PolicyODCreate object	策略属性

参数	参数类型	描述
operation_type	String	保护类型：备份（backup）、复制（replication）。 枚举值： <ul style="list-style-type: none">• backup• replication
trigger	PolicyTrigger Resp object	策略时间调度规则
associated_vaults	Array of PolicyAssociateVault objects	关联的存储库

表 4-356 PolicyoODCreate

参数	参数类型	描述
day_backups	Integer	保留日备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
destination_project_id	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。
destination_region	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。
max_backups	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100

参数	参数类型	描述
retention_duration_days	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups 参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	String	用户所在时区,格式形如UTC+08:00, 若选择年备，月备，周备，日备中任一参数，则该参数不能为空。
week_backups	Integer	保留周备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。
year_backups	Integer	保留年备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
full_backup_interval	Integer	每间隔多少次执行一次全量备份，当取值为 -1 时，不执行全量备份 最小值：-1 最大值：100

表 4-357 PolicyTriggerResp

参数	参数类型	描述
id	String	调度器id
name	String	调度器名称
properties	PolicyTriggerPropertiesResponse object	调度器属性
type	String	调度器类型,目前只支持 time: 定时调度。 枚举值： • time

表 4-358 PolicyTriggerPropertiesResp

参数	参数类型	描述
pattern	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天（MO、TU、WE、TH、FR、SA、SU），BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为： 'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00'。例如：某个地区的时间为 UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为 'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。
start_time	String	调度器开始时间，例如："2020-01-08 09:59:49"

表 4-359 PolicyAssociateVault

参数	参数类型	描述
destination_vault_id	String	关联的远端存储库ID
vault_id	String	存储库ID

请求示例

创建一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/policies
{
  "policy": {
    "enabled": true,
    "name": "policy001",
    "operation_definition": {
      "day_backups": 0,
      "month_backups": 0,
      "retention_duration_days": 1,
      "timezone": "UTC+08:00",
      "week_backups": 0,
      "year_backups": 0
    },
    "operation_type": "backup",
    "trigger": {
      "properties": {
        "pattern": [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ]
      }
    }
  }
}
```

```
    }
}
```

响应示例

状态码： 200

OK

```
{
  "policy": {
    "name": "policy001",
    "enabled": true,
    "trigger": {
      "properties": {
        "pattern": [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ],
        "start_time": "2019-05-08T06:57:05.000+00:00"
      },
      "type": "time",
      "id": "d67269a6-5369-42d7-8150-5254bd446328",
      "name": "default"
    },
    "operation_definition": {
      "retention_duration_days": 1,
      "year_backups": 0,
      "day_backups": 0,
      "month_backups": 0,
      "week_backups": 0,
      "timezone": "UTC+08:00"
    },
    "operation_type": "backup",
    "id": "cbb3ce6f-3332-4e7c-b98e-77290d8471ff"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class CreatePolicySolution {

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

```

```
this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
CreatePolicyRequest request = new CreatePolicyRequest();
PolicyCreateReq body = new PolicyCreateReq();
List<String> listPropertiesPattern = new ArrayList<>();

listPropertiesPattern.add("FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00");
PolicyTriggerPropertiesReq propertiesTrigger = new PolicyTriggerPropertiesReq();
propertiesTrigger.withPattern(listPropertiesPattern);
PolicyTriggerReq triggerPolicy = new PolicyTriggerReq();
triggerPolicy.withProperties(propertiesTrigger);
PolicyoODCreate operationDefinitionPolicy = new PolicyoODCreate();
operationDefinitionPolicy.withDayBackups(0)
    .withMonthBackups(0)
    .withRetentionDurationDays(1)
    .withTimezone("UTC+08:00")
    .withWeekBackups(0)
    .withYearBackups(0);
PolicyCreate policybody = new PolicyCreate();
policybody.withEnabled(true)
    .withName("policy001")
    .withOperationDefinition(operationDefinitionPolicy)
    .withOperationType(PolicyCreate.OperationTypeEnum.fromValue("backup"))
    .withTrigger(triggerPolicy);
body.withPolicy(policybody);
request.withBody(body);
try {
    CreatePolicyResponse response = client.createPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = CreatePolicyRequest()
    listPatternProperties = [
        "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00"
    ]
    propertiesTrigger = PolicyTriggerPropertiesReq(
        pattern=listPatternProperties
    )
    triggerPolicy = PolicyTriggerReq(
        properties=propertiesTrigger
    )
    operationDefinitionPolicy = PolicyoODCreate(
        day_backups=0,
        month_backups=0,
        retention_duration_days=1,
        timezone="UTC+08:00",
        week_backups=0,
        year_backups=0
    )
    policybody = PolicyCreate(
        enabled=True,
        name="policy001",
        operation_definition=operationDefinitionPolicy,
        operation_type="backup",
        trigger=triggerPolicy
    )
    request.body = PolicyCreateReq(
        policy=policybody
    )
    response = client.create_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

创建一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreatePolicyRequest{}
var listPatternProperties = []string{
    "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00",
}
propertiesTrigger := &model.PolicyTriggerPropertiesReq{
    Pattern: listPatternProperties,
}
triggerPolicy := &model.PolicyTriggerReq{
    Properties: propertiesTrigger,
}
dayBackupsOperationDefinition:= int32(0)
monthBackupsOperationDefinition:= int32(0)
retentionDurationDaysOperationDefinition:= int32(1)
timezoneOperationDefinition:= "UTC+08:00"
weekBackupsOperationDefinition:= int32(0)
yearBackupsOperationDefinition:= int32(0)
operationDefinitionPolicy := &model.PolicyoOdCreate{
    DayBackups: &dayBackupsOperationDefinition,
    MonthBackups: &monthBackupsOperationDefinition,
    RetentionDurationDays: &retentionDurationDaysOperationDefinition,
    Timezone: &timezoneOperationDefinition,
    WeekBackups: &weekBackupsOperationDefinition,
    YearBackups: &yearBackupsOperationDefinition,
}
enabledPolicy:= true
policybody := &model.PolicyCreate{
    Enabled: &enabledPolicy,
    Name: "policy001",
    OperationDefinition: operationDefinitionPolicy,
    OperationType: model.GetPolicyCreateOperationTypeEnum().BACKUP,
    Trigger: triggerPolicy,
}
request.Body = &model.PolicyCreateReq{
    Policy: policybody,
}
response, err := client.CreatePolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.8.2 查询单个策略

功能介绍

查询单个策略

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/policies/{policy_id}

表 4-360 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略ID
project_id	是	String	项目ID

请求参数

表 4-361 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 缺省值： MIIDkgYJKoZIhvcNAQcCoIDgzCCXXXX...

响应参数

状态码： 200

表 4-362 响应 Body 参数

参数	参数类型	描述
policy	Policy object	创建响应

表 4-363 Policy

参数	参数类型	描述
enabled	Boolean	策略是否启用
id	String	策略ID
name	String	策略名称
operation_definition	PolicyoODCreate object	策略属性
operation_type	String	保护类型：备份（backup）、复制（replication）。 枚举值： <ul style="list-style-type: none">• backup• replication
trigger	PolicyTrigger Resp object	策略时间调度规则
associated_vaults	Array of PolicyAssociateVault objects	关联的存储库

表 4-364 PolicyoODCreate

参数	参数类型	描述
day_backups	Integer	保留日备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
destination_project_id	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。

参数	参数类型	描述
destination_region	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。
max_backups	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
retention_duration_days	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups 参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	String	用户所在时区,格式形如UTC+08:00, 若选择年备，月备，周备，日备中任一参数，则该参数不能为空。
week_backups	Integer	保留周备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。
year_backups	Integer	保留年备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
full_backup_interval	Integer	每间隔多少次执行一次全量备份，当取值为 -1 时，不执行全量备份 最小值：-1 最大值：100 最小值：-1 最大值：100

表 4-365 PolicyTriggerResp

参数	参数类型	描述
id	String	调度器id
name	String	调度器名称
properties	PolicyTrigger PropertiesResponse object	调度器属性
type	String	调度器类型,目前只支持 time: 定时调度。 枚举值: <ul style="list-style-type: none">• time

表 4-366 PolicyTriggerPropertiesResp

参数	参数类型	描述
pattern	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天（MO、TU、WE、TH、FR、SA、SU），BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为： 'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00'。例如：某个地区的时间为 UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为 'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。
start_time	String	调度器开始时间，例如："2020-01-08 09:59:49"

表 4-367 PolicyAssociateVault

参数	参数类型	描述
destination_vault_id	String	关联的远端存储库ID
vault_id	String	存储库ID

请求示例

查询单个策略。

```
GET https://{{endpoint}}/v3/{{project_id}}/policies/{{policy_id}}
```

响应示例

状态码： 200

OK

```
{  
    "policy": {  
        "name": "policy001",  
        "associated_vaults": [ ],  
        "enabled": true,  
        "trigger": {  
            "properties": {  
                "pattern": [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ],  
                "start_time": "2019-05-08T06:57:05.000+00:00"  
            },  
            "type": "time",  
            "id": "d67269a6-5369-42d7-8150-5254bd446328",  
            "name": "default"  
        },  
        "operation_definition": {  
            "retention_duration_days": 1,  
            "year_backups": 0,  
            "day_backups": 0,  
            "month_backups": 0,  
            "week_backups": 0,  
            "timezone": "UTC+08:00"  
        },  
        "operation_type": "backup",  
        "id": "ccb3ce6f-3332-4e7c-b98e-77290d8471ff"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowPolicySolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ShowPolicyRequest request = new ShowPolicyRequest();
try {
    ShowPolicyResponse response = client.showPolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \
        client = CbrClient.new_builder() \
            .with_credentials(credentials) \
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \
            .build()

    try:
        request = ShowPolicyRequest()
        response = client.show_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"
cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.8.3 查询策略列表

功能介绍

[查询策略列表](#)

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/policies

表 4-368 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-369 Query 参数

参数	是否必选	参数类型	描述
operation_type	否	String	策略类型：备份（ backup ）、 复制(replication) 枚举值： • backup • replication
vault_id	否	String	存储库ID

请求参数

表 4-370 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务 获取用户Token接口获取（响应 消息头中X-Subject-Token的 值）。

响应参数

状态码： 200

表 4-371 响应 Body 参数

参数	参数类型	描述
policies	Array of Policy objects	策略列表
count	Integer	策略总数

表 4-372 Policy

参数	参数类型	描述
enabled	Boolean	策略是否启用
id	String	策略ID
name	String	策略名称
operation_definition	PolicyoODCreate object	策略属性
operation_type	String	保护类型：备份（backup）、复制（replication）。 枚举值： <ul style="list-style-type: none">• backup• replication
trigger	PolicyTrigger Resp object	策略时间调度规则
associated_vaults	Array of PolicyAssociateVault objects	关联的存储库

表 4-373 PolicyoODCreate

参数	参数类型	描述
day_backups	Integer	保留日备份个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
destination_project_id	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。
destination_region	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。

参数	参数类型	描述
max_backups	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
retention_duration_days	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	String	用户所在时区,格式形如UTC+08:00, 若选择年备，月备，周备，日备中任一参数，则该参数不能为空。
week_backups	Integer	保留周备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。
year_backups	Integer	保留年备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
full_backup_interval	Integer	每间隔多少次执行一次全量备份，当取值为 -1 时，不执行全量备份 最小值：-1 最大值：100

表 4-374 PolicyTriggerResp

参数	参数类型	描述
id	String	调度器id
name	String	调度器名称

参数	参数类型	描述
properties	PolicyTrigger PropertiesRe sp object	调度器属性
type	String	调度器类型,目前只支持 time: 定时调度。 枚举值: <ul style="list-style-type: none">• time

表 4-375 PolicyTriggerPropertiesResp

参数	参数类型	描述
pattern	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天（MO、TU、WE、TH、FR、SA、SU），BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为： 'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU; ;BYHOUR=14;BYMINUTE=00'。例如：某个地区的时间为 UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为 'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。
start_time	String	调度器开始时间，例如："2020-01-08 09:59:49"

表 4-376 PolicyAssociateVault

参数	参数类型	描述
destination_v ault_id	String	关联的远端存储库ID
vault_id	String	存储库ID

请求示例

查询指定备份策略列表

```
GET https://{endpoint}/v3/{project_id}/policies
```

响应示例

状态码： 200

OK

```
{  
    "policies": [ {  
        "name": "policy001",  
        "associated_vaults": [ ],  
        "enabled": true,  
        "trigger": {  
            "properties": {  
                "pattern": [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ],  
                "start_time": "2019-05-08T06:57:05.000+00:00"  
            },  
            "type": "time",  
            "id": "d67269a6-5369-42d7-8150-5254bd446328",  
            "name": "default"  
        },  
        "operation_definition": {  
            "retention_duration_days": 1,  
            "year_backups": 0,  
            "day_backups": 0,  
            "month_backups": 0,  
            "week_backups": 0,  
            "timezone": "UTC+08:00"  
        },  
        "operation_type": "backup",  
        "id": "cbb3ce6f-3332-4e7c-b98e-77290d8471ff"  
    } ],  
    "count": 10  
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

```

```
CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
ListPoliciesRequest request = new ListPoliciesRequest();
request.withOperationType(ListPoliciesRequest.OperationTypeEnum.fromValue("<operation_type>"));
request.withVaultId("<vault_id>");
try {
    ListPoliciesResponse response = client.listPolicies(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListPoliciesRequest()
        request.operation_type = "<operation_type>"
        request.vault_id = "<vault_id>"
        response = client.list_policies(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.ListPoliciesRequest{}
    operationTypeRequest:= model.GetListPoliciesRequestOperationTypeEnum().<OPERATION_TYPE>
    request.OperationType = &operationTypeRequest
    vaultIdRequest:= "<vault_id>"
    request.VaultId = &vaultIdRequest
    response, err := client.ListPolicies(request)
    if err == nil {
        fmt.Printf("%#v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.8.4 修改策略

功能介绍

修改策略

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/policies/{policy_id}

表 4-377 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略ID
project_id	是	String	项目ID

请求参数

表 4-378 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-379 请求 Body 参数

参数	是否必选	参数类型	描述
policy	是	PolicyUpdate object	修改策略body

表 4-380 PolicyUpdate

参数	是否必选	参数类型	描述
enabled	否	Boolean	是否启用策略 缺省值: true
name	否	String	策略名称 最小长度: 1 最大长度: 64
operation_definition	否	PolicyODCreate object	调度参数

参数	是否必选	参数类型	描述
trigger	否	PolicyTrigger Req object	策略执行时间规则

表 4-381 PolicyODCreate

参数	是否必选	参数类型	描述
day_backups	否	Integer	保留日备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone 也必选。 最小值：0 最大值：100
destination_project_id	否	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。
destination_region	否	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	否	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。
max_backups	否	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	否	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone 也必选。 最小值：0 最大值：100

参数	是否必选	参数类型	描述
retention_duration_days	否	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	否	String	用户所在时区,格式形如UTC +08:00, 若选择年备, 月备, 周备, 日备中任一参数, 则该参数不能为空。
week_backups	否	Integer	保留周备个数, 该备份不受保留最大备份数限制。取值为0到100。若选择该参数, 则timezone 也必选。
year_backups	否	Integer	保留年备个数, 该备份不受保留最大备份数限制。取值为0到100。若选择该参数, 则timezone 也必选。 最小值：0 最大值：100
full_backup_interval	否	Integer	每间隔多少次执行一次全量备份, 当取值为 -1 时, 不执行全量备份 最小值：-1 最大值：100

表 4-382 PolicyTriggerReq

参数	是否必选	参数类型	描述
properties	是	PolicyTriggerPropertiesRequest object	调度器属性

表 4-383 PolicyTriggerPropertiesReq

参数	是否必选	参数类型	描述
pattern	是	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天(MO、TU、WE、TH、FR、SA、SU)，BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为：'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00'。例如：某个地区的北京时间为UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为：'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。

响应参数

状态码： 200

表 4-384 响应 Body 参数

参数	参数类型	描述
policy	Policy object	创建响应

表 4-385 Policy

参数	参数类型	描述
enabled	Boolean	策略是否启用
id	String	策略ID
name	String	策略名称
operation_definition	PolicyODCreate object	策略属性

参数	参数类型	描述
operation_type	String	保护类型：备份（backup）、复制（replication）。 枚举值： <ul style="list-style-type: none">• backup• replication
trigger	PolicyTrigger Resp object	策略时间调度规则
associated_vaults	Array of PolicyAssociateVault objects	关联的存储库

表 4-386 PolicyoODCreate

参数	参数类型	描述
day_backups	Integer	保留日备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
destination_project_id	String	复制的目标项目ID，仅在跨区域复制时才会使用并且必须指定。
destination_region	String	复制的目标区域，仅在跨区域复制时才会使用并且必须指定。长度限制：0- 255，只能由字母、数字、“_”、“-”组成
enable_acceleration	Boolean	跨区域复制时，是否启用加速从而缩减复制的时间，如果不指定，默认不启用加速。
max_backups	Integer	单个备份对象自动备份的最大备份数。取值为-1或0-99999。-1代表不按备份数清理。若该字段和retention_duration_days字段同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
month_backups	Integer	保留月备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100

参数	参数类型	描述
retention_duration_days	Integer	备份保留时长，单位天。最长支持99999天。-1代表不按时间清理。若该字段和max_backups 参数同时为空，备份会永久保留。 最小值：1 最大值：99999 缺省值：-1
timezone	String	用户所在时区,格式形如UTC+08:00, 若选择年备，月备，周备，日备中任一参数，则该参数不能为空。
week_backups	Integer	保留周备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。
year_backups	Integer	保留年备个数，该备份不受保留最大备份数限制。取值为0到100。若选择该参数，则timezone也必选。 最小值：0 最大值：100
full_backup_interval	Integer	每间隔多少次执行一次全量备份，当取值为 -1 时，不执行全量备份 最小值：-1 最大值：100

表 4-387 PolicyTriggerResp

参数	参数类型	描述
id	String	调度器id
name	String	调度器名称
properties	PolicyTriggerPropertiesResponse object	调度器属性
type	String	调度器类型,目前只支持 time: 定时调度。 枚举值： • time

表 4-388 PolicyTriggerPropertiesResp

参数	参数类型	描述
pattern	Array of strings	调度规则。限制24条规则。调度器的调度规则，可参照iCalendar RFC 2445规范中的事件规则，但仅支持FREQ、BYDAY、BYHOUR、BYMINUTE、INTERVAL等参数，其中FREQ仅支持WEEKLY和DAILY，BYDAY支持一周七天（MO、TU、WE、TH、FR、SA、SU），BYHOUR支持0-23小时，BYMINUTE支持0-59分钟，并且间隔不能小于一小时，一天最大24个时间点。例如，周一到周天，每天UTC时间的14:00调度，其规则为： 'FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00'。例如：某个地区的时间为 UTC+8，若在该地区每天14:00调度，则在14点基础上减8，其规则为 'FREQ=DAILY;INTERVAL=1;BYHOUR=6;BYMINUTE=00'。
start_time	String	调度器开始时间，例如："2020-01-08 09:59:49"

表 4-389 PolicyAssociateVault

参数	参数类型	描述
destination_vault_id	String	关联的远端存储库ID
vault_id	String	存储库ID

请求示例

修改一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
PUT https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/policies/cbb3ce6f-3332-4e7c-b98e-77290d8471ff

{
  "policy" : {
    "enabled" : true,
    "name" : "policy001",
    "operation_definition" : {
      "day_backups" : 0,
      "month_backups" : 0,
      "max_backups" : 1,
      "timezone" : "UTC+08:00",
      "week_backups" : 0,
      "year_backups" : 0
    },
    "trigger" : {
      "properties" : {
        "pattern" : [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ]
      }
    }
  }
}
```

```
    }
}
```

响应示例

状态码： 200

OK

```
{
  "policy": {
    "name": "policy001",
    "associated_vaults": [],
    "enabled": true,
    "trigger": {
      "properties": {
        "pattern": [ "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00" ],
        "start_time": "2019-05-08T06:57:05.000+00:00"
      },
      "type": "time",
      "id": "d67269a6-5369-42d7-8150-5254bd446328",
      "name": "default"
    },
    "operation_definition": {
      "max_backups": 1,
      "year_backups": 0,
      "day_backups": 0,
      "month_backups": 0,
      "week_backups": 0,
      "timezone": "UTC+08:00"
    },
    "operation_type": "backup",
    "id": "cbb3ce6f-3332-4e7c-b98e-77290d8471ff"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class UpdatePolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_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);

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
UpdatePolicyRequest request = new UpdatePolicyRequest();
PolicyUpdateReq body = new PolicyUpdateReq();
List<String> listPropertiesPattern = new ArrayList<>();

listPropertiesPattern.add("FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00");
PolicyTriggerPropertiesReq propertiesTrigger = new PolicyTriggerPropertiesReq();
propertiesTrigger.withPattern(listPropertiesPattern);
PolicyTriggerReq triggerPolicy = new PolicyTriggerReq();
triggerPolicy.withProperties(propertiesTrigger);
PolicyoODCreate operationDefinitionPolicy = new PolicyoODCreate();
operationDefinitionPolicy.withDayBackups(0)
    .withMaxBackups(1)
    .withMonthBackups(0)
    .withTimezone("UTC+08:00")
    .withWeekBackups(0)
    .withYearBackups(0);
PolicyUpdate policybody = new PolicyUpdate();
policybody.withEnabled(true)
    .withName("policy001")
    .withOperationDefinition(operationDefinitionPolicy)
    .withTrigger(triggerPolicy);
body.withPolicy(policybody);
request.withBody(body);
try {
    UpdatePolicyResponse response = client.updatePolicy(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatus());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

修改一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
sk = os.getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

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

try:
    request = UpdatePolicyRequest()
    listPatternProperties = [
        "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00"
    ]
    propertiesTrigger = PolicyTriggerPropertiesReq(
        pattern=listPatternProperties
    )
    triggerPolicy = PolicyTriggerReq(
        properties=propertiesTrigger
    )
    operationDefinitionPolicy = PolicyoODCreate(
        day_backups=0,
        max_backups=1,
        month_backups=0,
        timezone="UTC+08:00",
        week_backups=0,
        year_backups=0
    )
    policybody = PolicyUpdate(
        enabled=True,
        name="policy001",
        operation_definition=operationDefinitionPolicy,
        trigger=triggerPolicy
    )
    request.body = PolicyUpdateReq(
        policy=policybody
    )
    response = client.update_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

修改一个备份策略，周一到周天每天14:00执行备份，保留策略按备份保留时长，保留一天。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    "region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>").
        WithCredential(auth).
        Build())

request := &model.UpdatePolicyRequest{}
var listPatternProperties = []string{
    "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU;BYHOUR=14;BYMINUTE=00",
}
propertiesTrigger := &model.PolicyTriggerPropertiesReq{
    Pattern: listPatternProperties,
}
triggerPolicy := &model.PolicyTriggerReq{
    Properties: propertiesTrigger,
}
dayBackupsOperationDefinition:= int32(0)
maxBackupsOperationDefinition:= int32(1)
monthBackupsOperationDefinition:= int32(0)
timezoneOperationDefinition:= "UTC+08:00"
weekBackupsOperationDefinition:= int32(0)
yearBackupsOperationDefinition:= int32(0)
operationDefinitionPolicy := &model.PolicyoOdCreate{
    DayBackups: &dayBackupsOperationDefinition,
    MaxBackups: &maxBackupsOperationDefinition,
    MonthBackups: &monthBackupsOperationDefinition,
    Timezone: &timezoneOperationDefinition,
    WeekBackups: &weekBackupsOperationDefinition,
    YearBackups: &yearBackupsOperationDefinition,
}
enabledPolicy:= true
namePolicy:= "policy001"
policybody := &model.PolicyUpdate{
    Enabled: &enabledPolicy,
    Name: &namePolicy,
    OperationDefinition: operationDefinitionPolicy,
    Trigger: triggerPolicy,
}
request.Body = &model.PolicyUpdateReq{
    Policy: policybody,
}
response, err := client.UpdatePolicy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.8.5 删除策略

功能介绍

删除策略

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project_id}/policies/{policy_id}

表 4-390 路径参数

参数	是否必选	参数类型	描述
policy_id	是	String	策略ID
project_id	是	String	项目ID

请求参数

表 4-391 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

无

请求示例

删除指定的备份策略。

```
DELETE https://{endpoint}/v3/{project_id}/policies/{policy_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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class DeletePolicySolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        DeletePolicyRequest request = new DeletePolicyRequest();
        try {
            DeletePolicyResponse response = client.deletePolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatus());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeletePolicyRequest()
        response = client.delete_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"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeletePolicyRequest{}
    response, err := client.DeletePolicy(request)
```

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

更多

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

状态码

状态码	描述
204	No Content

错误码

请参见[错误码](#)。

4.9 计量

4.9.1 查询容量统计

功能介绍

查询容量统计

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/storage_usage

表 4-392 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 4-393 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询条数
offset	否	Integer	偏移值
resource_id	否	String	支持按照备份对象ID过滤
resource_type	否	String	支持按照备份对象类型过滤 枚举值： <ul style="list-style-type: none">• OS::Nova::Server• OS::Ironic::BareMetalServer

请求参数

无

响应参数

状态码： 200

表 4-394 响应 Body 参数

参数	参数类型	描述
resource_count	Integer	满足过滤条件的资源总条数。
storage_usage	Array of StorageUsage objects	容量统计。

表 4-395 StorageUsage

参数	参数类型	描述
backup_count	Integer	备份数量
backup_size	Integer	备份容量
resource_id	String	资源ID
resource_name	String	资源名称
resource_type	String	资源类型

参数	参数类型	描述
backup_size_multiaz	Integer	多AZ备份大小

请求示例

无

响应示例

状态码： 200

OK

```
{  
    "storage_usage": [ {  
        "resource_name": "ECS_name",  
        "resource_id": "89d466e5-b7f9-4e45-807a-d5f17f44fc84",  
        "backup_size": 0,  
        "backup_count": 1,  
        "resource_type": "OS::Nova::Server",  
        "backup_size_multiaz": 0  
    } ],  
    "resource_count": 1  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowStorageUsageSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)
```

```
.withRegion(CbrRegion.valueOf("<YOUR REGION>"))
.build();
ShowStorageUsageRequest request = new ShowStorageUsageRequest();
request.withLimit(<limit>);
request.withOffset(<offset>);
request.withResourceId("<resource_id>");

request.withResourceType(ShowStorageUsageRequest.ResourceTypeEnum.fromValue("<resource_type>"));
try {
    ShowStorageUsageResponse response = client.showStorageUsage(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowStorageUsageRequest()
        request.limit = <limit>
        request.offset = <offset>
        request.resource_id = "<resource_id>"
        request.resource_type = "<resource_type>"
        response = client.show_storage_usage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowStorageUsageRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    resourceIdRequest:= "<resource_id>"
    request.ResourceId = &resourceIdRequest
    resourceTypeRequest:= model.GetShowStorageUsageRequestResourceTypeEnum().<RESOURCE_TYPE>
    request.ResourceType = &resourceTypeRequest
    response, err := client.ShowStorageUsage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.10 运营

4.10.1 变更

功能介绍

订单更新，支付cbc订单后，调用该接口更新包周期产品订单信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project_id}/orders/{order_id}

表 4-396 路径参数

参数	是否必选	参数类型	描述
order_id	是	String	订单ID
project_id	是	String	项目ID

请求参数

表 4-397 请求 Body 参数

参数	是否必选	参数类型	描述
cloudServiceConsoleURL	否	String	云服务ConsoleURL。订单支付完成后，客户可以通过此URL跳转到云服务Console页面查看信息 最小长度：1 最大长度：512
productInfo	是	CbcProductInfoUpdate object	产品信息
resourceId	是	String	待变更的资源ID
promotion_info	否	String	购买折扣

表 4-398 CbcProductInfoUpdate

参数	是否必选	参数类型	描述
productId	是	String	产品标识，通过订购询价接口获得，长度限制：1-64，只能由字母、数字、“_”、“-”组成。 最小长度：1 最大长度：64
resourceSize	是	Integer	资源容量大小，取值范围： 10-10485760 最小值：10 最大值：10485760
resourceSizeMeasureId	否	Integer	资源容量度量标识，枚举值 17: GB 缺省值：17
resourceSpecCode	是	String	用户购买云服务产品的资源规格 Enum: [vault.backup.server.normal, vault.backup.turbo.normal, vault.backup.database.normal , vault.backup.volume.normal, vault.backup.rds.normal, vault.replication.server.normal , vault.hybrid.server.normal] 枚举值： <ul style="list-style-type: none">• vault.backup.server.normal• vault.backup.turbo.normal• vault.backup.database.normal• vault.backup.volume.normal• vault.backup.rds.normal• vault.replication.server.normal• vault.hybrid.server.normal

响应参数

状态码： 200

表 4-399 响应 Body 参数

参数	参数类型	描述
orderId	String	订单ID
retCode	String	变更状态码
retMsg	String	变更信息

请求示例

变更包周期存储库大小到 50 GB

```
https://{{endpoint}}/v3/{{project_id}}/orders/CS2305081601LIS2V

{
  "resourceId" : "9dd8f8db-a926-440b-9af5-81b643618898",
  "productInfo" : {
    "productId" : "00301-231147-0--0",
    "resourceSpecCode" : "vault.backup.server.normal",
    "resourceSize" : 50,
    "resourceSizeMeasureId" : 17
  }
}
```

响应示例

状态码： 200

OK

```
{
  "orderId" : "CS2305291628MSG5A",
  "retCode" : 0,
  "retMsg" : "success"
}
```

SDK 代码示例

SDK 代码示例如下。

Java

变更包周期存储库大小到 50 GB

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class UpdateOrderSolution {

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

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.  
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_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);  
  
CbrClient client = CbrClient.newBuilder()  
.withCredential(auth)  
.withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
.build();  
UpdateOrderRequest request = new UpdateOrderRequest();  
CbcUpdate body = new CbcUpdate();  
CbcProductInfoUpdate productInfoBody = new CbcProductInfoUpdate();  
productInfoBody.withProductId("00301-231147-0--0")  
.withResourceSize(50)  
.withResourceSizeMeasureId(17)  
.withResourceSpecCode("vault.backup.server.normal");  
body.withResourceId("9dd8f8db-a926-440b-9af5-81b643618898");  
body.withProductInfo(productInfoBody);  
request.withBody(body);  
try {  
    UpdateOrderResponse response = client.updateOrder(request);  
    System.out.println(response.toString());  
} catch (ConnectionException e) {  
    e.printStackTrace();  
} catch (RequestTimeoutException e) {  
    e.printStackTrace();  
} catch (ServiceResponseException e) {  
    e.printStackTrace();  
    System.out.println(e.getHttpStatus());  
    System.out.println(e.getRequestId());  
    System.out.println(e.getErrorCode());  
    System.out.println(e.getErrorMsg());  
}  
}
```

Python

变更包周期存储库大小到 50 GB

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CbrClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CbrRegion.value_of("<YOUR REGION>")) \  
        .build()
```

```
try:
    request = UpdateOrderRequest()
    productInfoBody = CbcProductInfoUpdate(
        product_id="00301-231147-0--0",
        resource_size=50,
        resource_size_measure_id=17,
        resource_spec_code="vault.backup.server.normal"
    )
    request.body = CbcUpdate(
        resource_id="9dd8f8db-a926-440b-9af5-81b643618898",
        product_info=productInfoBody
    )
    response = client.update_order(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

变更包周期存储库大小到 50 GB

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateOrderRequest{}
    resourceSizeMeasureIdProductInfo := int32(17)
    productInfoBody := &model.CbcProductInfoUpdate{
        ProductId: "00301-231147-0--0",
        ResourceSize: int32(50),
        ResourceSizeMeasureId: &resourceSizeMeasureIdProductInfo,
        ResourceSpecCode: "vault.backup.server.normal",
    }
    request.Body = &model.CbcUpdate{
        ResourceId: "9dd8f8db-a926-440b-9af5-81b643618898",
        ProductInfo: productInfoBody,
    }
    response, err := client.UpdateOrder(request)
    if err == nil {
```

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.11 还原点

4.11.1 同步备份还原点

功能介绍

针对vault同步备份副本

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/checkpoints/sync

表 4-400 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-401 请求 Body 参数

参数	是否必选	参数类型	描述
sync	是	SyncParam object	同步内容

表 4-402 SyncParam

参数	是否必选	参数类型	描述
auto_trigger	是	Boolean	本次同步是否自动触发
vault_id	是	String	混合云 vault id

响应参数

状态码： 200

表 4-403 响应 Body 参数

参数	参数类型	描述
sync	SyncResponseBody object	同步内容

表 4-404 SyncResponseBody

参数	参数类型	描述
operation_log_id	String	是否自动触发
vault_id	String	混合云 vault id

请求示例

手动同步备份副本。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/checkpoints/sync
{
  "sync" : {
    "vault_id" : "8d3a8b28-a697-41e2-8d5d-903669f8a02d",
    "auto_trigger" : false
  }
}
```

响应示例

状态码： 200

OK

```
{  
    "sync": {  
        "operation_log_id": "60baed87-7e30-4bd8-b71c-89e4a4f7547d",  
        "vault_id": "8d3a8b28-a697-41e2-8d5d-903669f8a02d"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

手动同步备份副本。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ImportCheckpointSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ImportCheckpointRequest request = new ImportCheckpointRequest();  
        SyncReq body = new SyncReq();  
        SyncParam syncbody = new SyncParam();  
        syncbody.withAutoTrigger(false)  
            .withVaultId("8d3a8b28-a697-41e2-8d5d-903669f8a02d");  
        body.withSync(syncbody);  
        request.withBody(body);  
        try {  
            ImportCheckpointResponse response = client.importCheckpoint(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

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

Python

手动同步备份副本。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ImportCheckpointRequest()
        syncbody = SyncParam(
            auto_trigger=False,
            vault_id="8d3a8b28-a697-41e2-8d5d-903669f8a02d"
        )
        request.body = SyncReq(
            sync=syncbody
        )
        response = client.import_checkpoint(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

手动同步备份副本。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
    cbr.CbrClientBuilder().  
        WithRegion(region.ValueOf("<YOUR REGION>")).  
        WithCredential(auth).  
        Build())  
  
request := &model.ImportCheckpointRequest{}  
syncbody := &model.SyncParam{  
    AutoTrigger: false,  
    VaultId: "8d3a8b28-a697-41e2-8d5d-903669f8a02d",  
}  
request.Body = &model.SyncReq{  
    Sync: syncbody,  
}  
response, err := client.ImportCheckpoint(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.11.2 查询备份还原点

功能介绍

根据还原点ID查询指定还原点

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/checkpoints/{checkpoint_id}

表 4-405 路径参数

参数	是否必选	参数类型	描述
checkpoint_id	是	String	还原点ID
project_id	是	String	项目ID

请求参数

表 4-406 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 4-407 响应 Body 参数

参数	参数类型	描述
checkpoint	CheckpointCreate object	还原点信息

表 4-408 CheckpointCreate

参数	参数类型	描述
created_at	String	创建时间，例如："2020-02-05T10:38:34.209782"
id	String	还原点ID
project_id	String	项目ID

参数	参数类型	描述
status	String	状态:available,deleting,protecting,deleted,error-deleting,error 枚举值: <ul style="list-style-type: none">• available• deleting• protecting• deleted• error-deleting• error
vault	CheckpointPlanCreate object	存储库信息
extra_info	CheckpointExtraInfoResp object	扩展信息

表 4-409 CheckpointPlanCreate

参数	参数类型	描述
id	String	存储库id
name	String	存储库名称
resources	Array of CheckpointResourceResp objects	备份对象
skipped_resources	Array of CheckpointCreateSkippedResource objects	备份时跳过的资源列表

表 4-410 CheckpointResourceResp

参数	参数类型	描述
extra_info	String	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称

参数	参数类型	描述
protect_status	String	保护状态。available（可用），error（错误），protecting（备份中），restoring（恢复中），removing（删除中）。 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
resource_size	String	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	String	副本大小
backup_count	String	副本数量

表 4-411 CheckpointCreateSkippedResource

参数	参数类型	描述
id	String	资源ID
type	String	资源类型
name	String	资源名称
code	String	请参见 错误码 。
reason	String	跳过原因，例如：该资源正在备份中。

表 4-412 CheckpointExtraInfoResp

参数	参数类型	描述
name	String	备份名称
description	String	备份描述
retention_duration	Integer	备份保留天数

请求示例

GET https://[endpoint]/v3/4229d7a45436489f8c3dc2b1d35d4987/checkpoints/8b0851a8-adf3-4f4c-a914-dead08bf9664

响应示例

状态码： 200

OK

```
{  
    "checkpoint": {  
        "status": "available",  
        "created_at": "2019-05-10T07:59:12.037+00:00",  
        "vault": {  
            "id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",  
            "resources": [  
                {  
                    "name": "ecs-1f0f-0002",  
                    "resource_size": 40,  
                    "protect_status": "available",  
                    "type": "OS::Nova::Server",  
                    "id": "94eba8b2-acc9-4d82-badc-127144cc5526"  
                }],  
                "name": "vault-be94"  
            },  
            "project_id": "4229d7a45436489f8c3dc2b1d35d4987",  
            "id": "8b0851a8-adf3-4f4c-a914-dead08bf9664"  
        }  
    }  
}
```

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.11.3 备份还原点

功能介绍

执行复制

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/checkpoints/replicate

表 4-413 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-414 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-415 请求 Body 参数

参数	是否必选	参数类型	描述
replicate	是	CheckpointReplicateParam object	执行复制请求体参数

表 4-416 CheckpointReplicateParam

参数	是否必选	参数类型	描述
auto_trigger	否	Boolean	本次复制是否自动触发， 默认为 false， 代表手动触发
destination_project_id	是	String	复制的目标项目ID
destination_region	是	String	复制的目标区域id 最小长度： 0 最大长度： 255
destination_vault_id	是	String	目标区域存储库ID
enable_acceleration	否	Boolean	跨区域复制时， 是否启用加速从而缩短复制的时间，如果不指定， 默认不启用加速，如果启用加速，会额外收取加速的费用。
vault_id	是	String	存储库ID: uuid

响应参数

状态码： 200

表 4-417 响应 Body 参数

参数	参数类型	描述
replication	CheckpointReplicateRespBody object	复制备份还原点响应对象。

表 4-418 CheckpointReplicateRespBody

参数	参数类型	描述
backups	Array of CheckpointReplicateRespBackups objects	待复制的备份列表
destination_project_id	String	复制的目标项目ID
destination_region	String	复制的目标区域
destination_vault_id	String	目标区域存储库ID
project_id	String	执行复制的项目ID
provider_id	String	备份提供商ID
source_region	String	复制的源区域
vault_id	String	存储库ID

表 4-419 CheckpointReplicateRespbackups

参数	参数类型	描述
backup_id	String	待复制的备份ID
replication_record_id	String	复制记录ID

请求示例

手动跨区域复制一个备份副本从源区域到目标区域。

```
POST https://{endpoint}/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/checkpoints/replicate
```

```
{  
    "replicate": {  
        "auto_trigger": false,  
        "destination_project_id": "68589cac08274b82b4e254268a3862d8",  
        "destination_region": "region2",  
        "destination_vault_id": "0ca3eb86-8800-46da-9c37-9d657a825274",  
        "enable_acceleration": false,  
        "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce"  
    }  
}
```

响应示例

状态码： 200

OK

```
{  
    "replication": {  
        "destination_vault_id": "0ca3eb86-8800-46da-9c37-9d657a825274",  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "source_region": "region1",  
        "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",  
        "destination_region": "region2",  
        "destination_project_id": "68589cac08274b82b4e254268a3862d8",  
        "backups": [ {  
            "replication_record_id": "de128dfa-5451-4905-9c11-8fc842b2f41e",  
            "backup_id": "7558e1a3-7046-4787-95cd-14b0ad0642a8"  
        }, {  
            "replication_record_id": "892a7d1e-17c8-4751-ad75-cfbed7051857",  
            "backup_id": "6df2b54c-dd62-4059-a07c-1b8f24f2725d"  
        }, {  
            "replication_record_id": "8bf5ce8f-bfa1-4d57-98de-d6159ab9d86d",  
            "backup_id": "aa00034d-ef40-443d-ab7a-dc846d988cdf"  
        } ],  
        "project_id": "4229d7a45436489f8c3dc2b1d35d4987"  
    }  
}
```

SDK 代码示例

SDK代码示例如下。

Java

手动跨区域复制一个备份副本从源区域到目标区域。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class CopyCheckpointSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    }  
}
```

```
environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running  
    this example, set environment variables CLOUD_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);  
  
    CbrClient client = CbrClient.newBuilder()  
        .withCredential(auth)  
        .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
        .build();  
    CopyCheckpointRequest request = new CopyCheckpointRequest();  
    CheckpointReplicateReq body = new CheckpointReplicateReq();  
    CheckpointReplicateParam replicatebody = new CheckpointReplicateParam();  
    replicatebody.withAutoTrigger(false)  
        .withDestinationProjectId("68589cac08274b82b4e254268a3862d8")  
        .withDestinationRegion("region2")  
        .withDestinationVaultId("0ca3eb86-8800-46da-9c37-9d657a825274")  
        .withEnableAcceleration(false)  
        .withVaultId("3b5816b5-f29c-4172-9d9a-76c719a659ce");  
    body.withReplicate(replicatebody);  
    request.withBody(body);  
    try {  
        CopyCheckpointResponse response = client.copyCheckpoint(request);  
        System.out.println(response.toString());  
    } catch (ConnectionException e) {  
        e.printStackTrace();  
    } catch (RequestTimeoutException e) {  
        e.printStackTrace();  
    } catch (ServiceResponseException e) {  
        e.printStackTrace();  
        System.out.println(e.getHttpStatus());  
        System.out.println(e.getRequestId());  
        System.out.println(e.getErrorCode());  
        System.out.println(e.getErrorMsg());  
    }  
}
```

Python

手动跨区域复制一个备份副本从源区域到目标区域。

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")  
    sk = os.getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = CbrClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CbrRegion.value_of("<YOUR REGION>")) \  
        .build()
```

```
try:
    request = CopyCheckpointRequest()
    replicatebody = CheckpointReplicateParam(
        auto_trigger=False,
        destination_project_id="68589cac08274b82b4e254268a3862d8",
        destination_region="region2",
        destination_vault_id="0ca3eb86-8800-46da-9c37-9d657a825274",
        enable_acceleration=False,
        vault_id="3b5816b5-f29c-4172-9d9a-76c719a659ce"
    )
    request.body = CheckpointReplicateReq(
        replicate=replicatebody
    )
    response = client.copy_checkpoint(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

手动跨区域复制一个备份副本从源区域到目标区域。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CopyCheckpointRequest{}
    autoTriggerReplicate:= false
    enableAccelerationReplicate:= false
    replicatebody := &model.CheckpointReplicateParam{
        AutoTrigger: &autoTriggerReplicate,
        DestinationProjectId: "68589cac08274b82b4e254268a3862d8",
        DestinationRegion: "region2",
        DestinationVaultId: "0ca3eb86-8800-46da-9c37-9d657a825274",
        EnableAcceleration: &enableAccelerationReplicate,
        VaultId: "3b5816b5-f29c-4172-9d9a-76c719a659ce",
    }
    request.Body = &model.CheckpointReplicateReq{
        Replicate: replicatebody,
    }
```

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.11.4 创建备份还原点

功能介绍

对存储库执行备份，生成备份还原点

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project_id}/checkpoints

表 4-420 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 4-421 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 4-422 请求 Body 参数

参数	是否必选	参数类型	描述
checkpoint	是	VaultBackup object	存储库执行备份参数

表 4-423 VaultBackup

参数	是否必选	参数类型	描述
parameters	否	CheckpointParam object	参数
vault_id	是	String	存储库ID

表 4-424 CheckpointParam

参数	是否必选	参数类型	描述
auto_trigger	否	Boolean	是否自动触发,true:自动触发, false: 非自动触发。 缺省值: false
description	否	String	备份描述 最小长度: 0 最大长度: 255
incremental	否	Boolean	是否增量备份, true:增量备份, false: 非增量备份。 缺省值: true

参数	是否必选	参数类型	描述
name	否	String	备份名称，只能由数字、中文、英文大小写字母，以及下划线 "_" 和中横线 "-" 字符构成。 最小长度：1 最大长度：64
resources	否	Array of strings	待备份的资源id列表:uuid
resource_details	否	Array of Resource objects	资源详情 数组长度：0 - 256

表 4-425 Resource

参数	是否必选	参数类型	描述
extra_info	否	ResourceExtraInfo object	资源附加信息
id	是	String	待备份资源id
name	否	String	待备份资源名称，长度限制： 0-255 最小长度：0 最大长度：255
type	是	String	待备份资源的类型： OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2

表 4-426 ResourceExtraInfo

参数	是否必选	参数类型	描述
exclude_volumes	否	Array of strings	需要排除备份的卷id。仅在多盘备份特性中有效，排除不需要备份的磁盘。当虚拟机新绑定磁盘时，也能继续排除之前设置不用备份的卷。

响应参数

状态码： 200

表 4-427 响应 Body 参数

参数	参数类型	描述
checkpoint	CheckpointCreate object	还原点信息

表 4-428 CheckpointCreate

参数	参数类型	描述
created_at	String	创建时间，例如："2020-02-05T10:38:34.209782"
id	String	还原点ID
project_id	String	项目ID
status	String	状态:available,deleting,protecting,deleted,error-deleting,error 枚举值： <ul style="list-style-type: none">• available• deleting• protecting• deleted• error-deleting• error
vault	CheckpointPlanCreate object	存储库信息
extra_info	CheckpointExtraInfoResp object	扩展信息

表 4-429 CheckpointPlanCreate

参数	参数类型	描述
id	String	存储库id
name	String	存储库名称

参数	参数类型	描述
resources	Array of CheckpointResourceResp objects	备份对象
skipped_resources	Array of CheckpointCreateSkippedResource objects	备份时跳过的资源列表

表 4-430 CheckpointResourceResp

参数	参数类型	描述
extra_info	String	资源附加信息
id	String	待备份资源id
name	String	待备份资源名称
protect_status	String	保护状态。available（可用），error（错误），protecting（备份中），restoring（恢复中），removing（删除中）。 枚举值： <ul style="list-style-type: none">• available• error• protecting• restoring• removing
resource_size	String	资源已分配容量,单位为GB
type	String	待备份资源的类型: OS::Nova::Server, OS::Cinder::Volume, OS::Ironic::BareMetalServer, OS::Native::Server, OS::Sfs::Turbo, OS::Workspace::DesktopV2
backup_size	String	副本大小
backup_count	String	副本数量

表 4-431 CheckpointCreateSkippedResource

参数	参数类型	描述
id	String	资源ID

参数	参数类型	描述
type	String	资源类型
name	String	资源名称
code	String	请参见 错误码 。
reason	String	跳过原因，例如：该资源正在备份中。

表 4-432 CheckpointExtraInfoResp

参数	参数类型	描述
name	String	备份名称
description	String	备份描述
retention_duration	Integer	备份保留天数

请求示例

执行存储库资源自动备份，生成备份还原点。

```
POST https://[endpoint]/v3/f841e01fd2b14e7fa41b6ae7aa6b0594/checkpoints

{
  "checkpoint": {
    "parameters": {
      "auto_trigger": true,
      "description": "backup_description",
      "incremental": true,
      "name": "backup_name",
      "resources": [ "94eba8b2-acc9-4d82-badc-127144cc5526" ]
    },
    "vault_id": "3b5816b5-f29c-4172-9d9a-76c719a659ce"
  }
}
```

响应示例

状态码： 200

OK

```
{
  "checkpoint": {
    "status": "protecting",
    "created_at": "2019-05-10T07:59:12.733+00:00",
    "vault": {
      "skipped_resources": [ ],
      "id": "3b5816b5-f29c-4172-9d9a-76c719a659ce",
      "resources": [ {
        "name": "ecs-1f0f-0002",
        "resource_size": 40,
        "protect_status": "available",
        "type": "OS::Nova::Server",
        "id": "94eba8b2-acc9-4d82-badc-127144cc5526"
      } ]
    }
  }
}
```

```
        },
        "name" : "vault-be94"
    },
    "project_id" : "4229d7a45436489f8c3dc2b1d35d4987",
    "id" : "8b0851a8-adf3-4f4c-a914-dead08bf9664",
    "extra_info" : {
        "retention_duration" : -1,
        "name" : "backup_name",
        "description" : "backup_description"
    }
}
```

SDK 代码示例

SDK代码示例如下。

Java

执行存储库资源自动备份，生成备份还原点。

```
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.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

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

public class CreateCheckpointSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateCheckpointRequest request = new CreateCheckpointRequest();
        VaultBackupReq body = new VaultBackupReq();
        List<String> listParametersResources = new ArrayList<>();
        listParametersResources.add("94eba8b2-acc9-4d82-badc-127144cc5526");
        CheckpointParam parametersCheckpoint = new CheckpointParam();
        parametersCheckpoint.withAutoTrigger(true)
            .withDescription("backup_description")
            .withIncremental(true)
            .withName("backup_name")
            .withResources(listParametersResources);
        VaultBackup checkpointbody = new VaultBackup();
        checkpointbody.withParameters(parametersCheckpoint)
            .withVaultId("3b5816b5-f29c-4172-9d9a-76c719a659ce");
```

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

Python

执行存储库资源自动备份，生成备份还原点。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateCheckpointRequest()
        listResourcesParameters = [
            "94eba8b2-acc9-4d82-badc-127144cc5526"
        ]
        parametersCheckpoint = CheckpointParam(
            auto_trigger=True,
            description="backup_description",
            incremental=True,
            name="backup_name",
            resources=listResourcesParameters
        )
        checkpointbody = VaultBackup(
            parameters=parametersCheckpoint,
            vault_id="3b5816b5-f29c-4172-9d9a-76c719a659ce"
        )
        request.body = VaultBackupReq(
            checkpoint=checkpointbody
        )
        response = client.create_checkpoint(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

Go

执行存储库资源自动备份，生成备份还原点。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateCheckpointRequest{}
    var listResourcesParameters = []string{
        "94eba8b2-acc9-4d82-badc-127144cc5526",
    }
    autoTriggerParameters:= true
    descriptionParameters:= "backup_description"
    incrementalParameters:= true
    nameParameters:= "backup_name"
    parametersCheckpoint := &model.CheckpointParam{
        AutoTrigger: &autoTriggerParameters,
        Description: &descriptionParameters,
        Incremental: &incrementalParameters,
        Name: &nameParameters,
        Resources: &listResourcesParameters,
    }
    checkpointbody := &model.VaultBackup{
        Parameters: parametersCheckpoint,
        VaultId: "3b5816b5-f29c-4172-9d9a-76c719a659ce",
    }
    request.Body = &model.VaultBackupReq{
        Checkpoint: checkpointbody,
    }
    response, err := client.CreateCheckpoint(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.12 项目

4.12.1 查询租户的项目信息

功能介绍

查询租户的企业项目信息

接口约束

本接口仅用于开通了企业项目场景中，Console需要获取复制目标区域的project_id时使用，其它场景下调用无返回值。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/region-projects

请求参数

无

响应参数

状态码： 200

表 4-433 响应 Body 参数

参数	参数类型	描述
projects	Array of ProjectsListInfo objects	项目信息
links	SelfLinksInfo object	连接地址

表 4-434 ProjectsListInfo

参数	参数类型	描述
domain_id	String	域 ID
is_domain	Boolean	是否是域级
parent_id	String	父项目 ID
name	String	名称
description	String	描述信息
id	String	项目ID
enabled	Boolean	是否开启
links	SelfLinksInfo object	连接地址

表 4-435 SelfLinksInfo

参数	参数类型	描述
self	String	连接地址

请求示例

```
GET https://{endpoint}/v3/region-projects
```

响应示例

状态码： 200

OK

```
{  
  "projects" : [ {  
    "description" : "",  
    "domain_id" : "5ba348512fd541c3a57c9bb2196108f8",  
    "enabled" : true,  
    "id" : "45a3afcc926f47d08b9375694b062193",  
    "links" : {  
      "self" : "https://.../v3/region-projects/45a3afcc926f47d08b9375694b062193"  
    }  
  } ]  
}
```

```
"is_domain" : false,
"links" : {
    "self" : "https://iam.huaweicloud.com:31943/v3/projects/45a3afcc926f47d08b9375694b062193"
},
"name" : "cn-hk1_yyx2",
"parent_id" : ""
}, {
"description" : "",
"domain_id" : "5ba348512fd541c3a57c9bb2196108f8",
"enabled" : true,
"id" : "438fe051ccb04aeabc0c8112a276a0dc",
"is_domain" : false,
"links" : {
    "self" : "https://iam.huaweicloud.com:31943/v3/projects/438fe051ccb04aeabc0c8112a276a0dc"
},
"name" : "cn-xianhz-1",
"parent_id" : ""
} ],
"links" : {
    "self" : "https://iam.huaweicloud.com:31943/v3/auth/projects"
}
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ListProjectsSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ListProjectsRequest request = new ListProjectsRequest();
        try {
            ListProjectsResponse response = client.listProjects(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

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

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
    cbr.CbrClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.12.2 查询租户项目列表

功能介绍

根据指定租户名称查询项目列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/domain/{domain_name}/projects

表 4-436 路径参数

参数	是否必选	参数类型	描述
domain_name	是	String	租户名称

请求参数

无

响应参数

状态码： 200

表 4-437 响应 Body 参数

参数	参数类型	描述
projects	Array of DomainProjectsInfo objects	项目列表

表 4-438 DomainProjectsInfo

参数	参数类型	描述
project_id	String	项目ID
project_name	String	项目名称

请求示例

```
GET https://[endpoint]/v3/domain/{domain_name}/projects
```

响应示例

状态码： 200

OK

```
{
  "projects": [
    {
      "project_name": "cn-north-7",
      "project_id": "fe1431a61f4242dd87e548f8f42ec0b4"
    },
    {
      "project_name": "cn-north-7-virtual",
      "project_id": "bccd81411c254e6890fe88d2f2ce406b"
    }
  ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ListDomainProjectsSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ListDomainProjectsRequest request = new ListDomainProjectsRequest();
        try {
            ListDomainProjectsResponse response = client.listDomainProjects(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.v1 import *

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

    credentials = BasicCredentials(ak, sk) \\\
        client = CbrClient.new_builder() \\
            .with_credentials(credentials) \\
            .with_region(CbrRegion.value_of("<YOUR REGION>")) \\
            .build()
```

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

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(  
        cbr.CbrClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.ListDomainProjectsRequest{}  
    response, err := client.ListDomainProjects(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.12.3 查询租户信息

功能介绍

由控制台调用的内部接口，用于仅在查询共享备份时获取源project_id的域名信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/domain/{source_project_id}

表 4-439 路径参数

参数	是否必选	参数类型	描述
source_project_id	是	String	源项目ID

请求参数

无

响应参数

状态码： 200

表 4-440 响应 Body 参数

参数	参数类型	描述
project_name	String	项目名称。
project_id	String	项目ID。
domain_id	String	项目所属帐号ID。
domain_name	String	项目所属帐号名称。

请求示例

无

响应示例

状态码： 200

OK

```
{  
    "project_id": "fe1431a61f4242dd87e548f8f42ec0b4",  
    "project_name": "cn-north-7",  
    "domain_id": "domain_id",  
    "domain_name": "domain_name"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;  
import com.huaweicloud.sdk.cbr.v1.*;  
import com.huaweicloud.sdk.cbr.v1.model.*;  
  
public class ShowDomainSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_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);  
  
        CbrClient client = CbrClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowDomainRequest request = new ShowDomainRequest();  
        try {  
            ShowDomainResponse response = client.showDomain(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDomainRequest{}
    response, err := client.ShowDomain(request)
```

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

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.12.4 租户迁移

功能介绍

将CSBS/VBS资源迁移到CBR。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/migrates

请求参数

表 4-441 请求 Body 参数

参数	是否必选	参数类型	描述
all_regions	是	Boolean	是否触发其他区域迁移 缺省值: true
reservation	是	Float	存储库默认扩容比，取值范围0到1 最小值: 0 最大值: 1 缺省值: 0.2

响应参数

状态码： 200

表 4-442 响应 Body 参数

参数	参数类型	描述
{自定义key}	Map<String, String>	OK

请求示例

将CSBS/VBS资源迁移到CBR。

```
POST https://{endpoint}/v3/migrates
```

```
{
  "all_regions" : true,
  "reservation" : 0.2
}
```

响应示例

状态码： 200

OK

```
{
  "cn-north-7" : "running"
}
```

SDK 代码示例

SDK代码示例如下。

Java

将CSBS/VBS资源迁移到CBR。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class MigrateDomainSolution {

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

CbrClient client = CbrClient.newBuilder()
    .withCredential(auth)
    .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
    .build();
MigrateDomainRequest request = new MigrateDomainRequest();
DomainMigrate body = new DomainMigrate();
body.withReservation(0.2f);
body.withAllRegions(true);
request.withBody(body);
try {
    MigrateDomainResponse response = client.migrateDomain(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException 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CSBS/VBS资源迁移到CBR。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = MigrateDomainRequest()
        request.body = DomainMigrate(
            reservation=0.2,
            all_regions=True
        )
        response = client.migrate_domain(request)
        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CSBS/VBS资源迁移到CBR。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/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 := cbr.NewCbrClient(
        cbr.CbrClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>").
            WithCredential(auth).
            Build()))

    request := &model.MigrateDomainRequest{}
    request.Body = &model.DomainMigrate{
        Reservation: float32(0.2),
        AllRegions: true,
    }
    response, err := client.MigrateDomain(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

4.12.5 查询迁移

功能介绍

查询迁移结果

调用方法

请参见[如何调用API](#)。

URI

GET /v3/migrates

表 4-443 Query 参数

参数	是否必选	参数类型	描述
all_regions	否	Boolean	是否查询其他区域结果 缺省值: true

请求参数

无

响应参数

状态码: 200

表 4-444 响应 Body 参数

参数	参数类型	描述
status	String	租户迁移状态 枚举值: <ul style="list-style-type: none">● success● failed● migrating
project_status	Array of DomainMigrateProjectStatus objects	项目迁移状态

表 4-445 DomainMigrateProjectStatus

参数	参数类型	描述
status	String	迁移状态 枚举值： <ul style="list-style-type: none">• migrating• success• failed
project_id	String	项目ID
project_name	String	项目名称
region_id	String	区域ID
progress	Integer	迁移进度 最小值： 0 最大值： 100
fail_code	Integer	失败错误码（仅当项目状态为失败时才有该参数）。
fail_reason	String	失败原因（仅当项目状态为失败时才有该参数）。

请求示例

<https://{{endpoint}}/v3/migrates>

响应示例

状态码： 200

OK

```
{  
    "status" : "failed",  
    "project_status" : [ {  
        "status" : "failed",  
        "progress" : 1,  
        "project_id" : "fe1431a61f4242dd87e548f8f42ec0b4",  
        "project_name" : "cn-north-7",  
        "region_id" : "cn-north-7",  
        "fail_code" : 4002,  
        "fail_reason" : "An error occurred when converting models: dictionary keys changed during iteration."  
    }, {  
        "status" : "success",  
        "progress" : 100,  
        "project_id" : "bccd81411c254e6890fe88d2f2ce406b",  
        "project_name" : "cn-north-7-virtual",  
        "region_id" : "cn-north-7"  
    } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.cbr.v1.region.CbrRegion;
import com.huaweicloud.sdk.cbr.v1.*;
import com.huaweicloud.sdk.cbr.v1.model.*;

public class ShowMigrateStatusSolution {

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

        CbrClient client = CbrClient.newBuilder()
            .withCredential(auth)
            .withRegion(CbrRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowMigrateStatusRequest request = new ShowMigrateStatusRequest();
        request.withAllRegions(<all_regions>);
        try {
            ShowMigrateStatusResponse response = client.showMigrateStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcbr.v1.region.cbr_region import CbrRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcbr.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.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \n\nclient = CbrClient.new_builder() \n    .with_credentials(credentials) \n    .with_region(CbrRegion.value_of("<YOUR REGION>")) \n    .build()\n\ntry:\n    request = ShowMigrateStatusRequest()\n    request.all_regions = <AllRegions>\n    response = client.show_migrate_status(request)\n    print(response)\nexcept exceptions.ClientRequestException as e:\n    print(e.status_code)\n    print(e.request_id)\n    print(e.error_code)\n    print(e.error_msg)
```

Go

```
package main\n\nimport (\n    "fmt"\n    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"\n    cbr "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1"\n    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/model"\n    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/cbr/v1/region"\n)\n\nfunc main() {\n    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security\n    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment\n    // variables and decrypted during use to ensure security.\n    // In this example, AK and SK are stored in environment variables for authentication. Before running this\n    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment\n    ak := os.Getenv("CLOUD_SDK_AK")\n    sk := os.Getenv("CLOUD_SDK_SK")\n\n    auth := basic.NewCredentialsBuilder().\n        WithAk(ak).\n        WithSk(sk).\n        Build()\n\n    client := cbr.NewCbrClient(\n        cbr.CbrClientBuilder().\n            WithRegion(region.ValueOf("<YOUR REGION>")).\n            WithCredential(auth).\n            Build())\n\n    request := &model.ShowMigrateStatusRequest{}\n    allRegionsRequest:= <all_regions>\n    request.AllRegions = &allRegionsRequest\n    response, err := client.ShowMigrateStatus(request)\n    if err == nil {\n        fmt.Printf("%+v\n", response)\n    } else {\n        fmt.Println(err)\n    }\n}
```

更多

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

状态码

状态码	描述
200	OK

错误码

请参见[错误码](#)。

5 应用示例

5.1 示例 1：创建弹性云服务器备份

场景描述

本章节指导用户通过API创建云服务器备份。API的调用方法请参见[如何调用API](#)。

创建备份时，支持云服务器或磁盘等资源备份。本节以云服务器为例，介绍如何创建云服务器备份。

约束限制

该接口以创建按需计费的存储库为例。

涉及接口

创建云服务器备份时，需要进行创建备份的容器存储库，将服务器或磁盘等资源挂载至存储库，为服务器或磁盘等资源创建备份。涉及的接口如下：

- [创建存储库](#)：创建备份的容器。
- [为存储库添加保护资源](#)：确定待备份云服务器或磁盘。
- [创建备份还原点](#)：创建备份。
- [查询备份还原点](#)：确认备份创建成功。

操作步骤

1. 创建存储库。
 - a. 创建简单配置的存储库。
 - 接口相关信息
URI格式：POST /v3/{project_id}/vaults
详情请参见[创建存储库](#)。
 - 请求示例
POST: `https://{{endpoint}}/v3/{{project_id}}/vaults`

{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{  
    "vault": {  
        "billing": {  
            "cloud_type": "public",  
            "consistent_level": "crash_consistent",  
            "object_type": "server",  
            "protect_type": "backup",  
            "size": 200  
        },  
        "name": "my_vault",  
        "resources": []  
    }  
}
```

■ 响应示例

```
{  
    "vault": {  
        "id": "ea7b8717-2543-478a-a92d-3ca7ee448f67",  
        "name": "my_vault",  
        "description": null,  
        "resources": [],  
        "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
        "created_at": "2020-08-17T03:51:24.678916",  
        "project_id": "0605767b5780d5762fc5c0118072a564",  
        "enterprise_project_id": "0",  
        "auto_bind": false,  
        "bind_rules": {},  
        "user_id": "aa2999fa5ae640f28926f8fd79188934",  
        "billing": {  
            "allocated": 0,  
            "cloud_type": "public",  
            "consistent_level": "crash_consistent",  
            "frozen_scene": null,  
            "charging_mode": "post_paid",  
            "order_id": null,  
            "product_id": null,  
            "protect_type": "backup",  
            "object_type": "server",  
            "spec_code": "vault.backup.server.normal",  
            "used": 0,  
            "storage_unit": null,  
            "status": "available",  
            "size": 200  
        },  
        "tags": []  
    }  
}
```

b. 创建包周期存储库并直接绑定服务器。

■ 接口相关信息

URI格式: POST /v3/{project_id}/vaults

接口与[步骤a](#)保持一致。

■ 请求示例

POST: https://{endpoint}/v3/{project_id}/vaults

{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{  
    "vault": {  
        "billing": {  
            "cloud_type": "public",  
            "consistent_level": "crash_consistent",  
            "object_type": "server",  
            "protect_type": "backup",  
            "size": 200  
        },  
        "name": "my_vault",  
        "resources": []  
    }  
}
```

```
        "consistent_level": "crash_consistent",
        "object_type": "server",
        "protect_type": "backup",
        "size": 100,
        "charging_mode": "pre_paid",
        "period_type": "month",
        "period_num": 1,
        "is_auto_renew": false,
        "is_auto_pay": false
    },
    "description": "vault_description",
    "name": "vault_name",
    "resources": [
        {
            "id": "97595625-198e-4e4d-879b-9d53f68ba551",
            "type": "OS::Nova::Server"
        }
    ]
}
```

- c. 记录响应消息体中存储库ID。
2. 为存储库添加保护资源，将服务器/磁盘绑定到存储库中。
- a. 添加资源

▪ 接口相关信息

URI格式：POST /v3/{project_id}/vaults/{vault_id}/addresources
详情请参见[添加资源](#)。

▪ 请求示例

POST: [https://\[endpoint\]/v3/0605767b5780d5762fc5c0118072a564/vaults/ea7b8717-2543-478a-a92d-3ca7ee448f67/addresources](https://[endpoint]/v3/0605767b5780d5762fc5c0118072a564/vaults/ea7b8717-2543-478a-a92d-3ca7ee448f67/addresources)
{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{
    "resources": [
        {
            "id": "e8cc6bfd-d324-4b88-9109-9fb0ba70676f",
            "type": "OS::Nova::Server",
            "name": "server-4690-0002"
        }
    ]
}
```

▪ 响应示例

```
{
    "add_resource_ids": [
        "e8cc6bfd-d324-4b88-9109-9fb0ba70676f"
    ]
}
```

- b. 请求体中根据需要选择运行中且没有绑定存储库的云服务器ID。
3. 创建备份还原点。
- a. 创建备份还原点。

▪ 接口相关信息

URI格式：POST /v3/{project_id}/checkpoints
详情请参见[创建备份还原点](#)。

▪ 请求示例

POST: [https://\[endpoint\]/v3/0605767b5780d5762fc5c0118072a564/checkpoints](https://[endpoint]/v3/0605767b5780d5762fc5c0118072a564/checkpoints)
{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{  
    "checkpoint": {  
        "parameters": {  
            "auto_trigger": false,  
            "description": "backupauto",  
            "incremental": true,  
            "name": "backup_auto",  
            "resources": ["e8cc6bfd-d324-4b88-9109-9fb0ba70676f"]  
        },  
        "vault_id": "ea7b8717-2543-478a-a92d-3ca7ee448f67"  
    }  
}
```

■ 响应示例

```
{  
    "checkpoint": {  
        "id": "d9ce6924-d753-4132-bd16-a9f8838ea7d2",  
        "project_id": "0605767b5780d5762fc5c0118072a564",  
        "status": "protecting",  
        "vault": {  
            "id": "ea7b8717-2543-478a-a92d-3ca7ee448f67",  
            "name": "my_vault",  
            "resources": [  
                {  
                    "id": "e8cc6bfd-d324-4b88-9109-9fb0ba70676f",  
                    "type": "OS::Nova::Server",  
                    "name": "ecs-9f93-0002",  
                    "extra_info": "{}",  
                    "resource_size": "40",  
                    "backup_size": "0",  
                    "backup_count": "0",  
                    "protect_status": "available"  
                }  
            ],  
            "skipped_resources": []  
        },  
        "created_at": "2020-08-17T06:49:06.307378",  
        "extra_info": {  
            "name": "backup_auto",  
            "description": "backupauto",  
            "retention_duration": -1  
        }  
    }  
}
```

b. 记录响应消息体中备份还原点ID。

4. 确认服务器备份成功。**- 接口相关信息**

URI格式: GET /v3/{project_id}/checkpoints/{checkpoint_id}

详情请参见[查询备份还原点](#)。

{endpoint}信息请从[地区和终端节点](#)获取。

- 请求示例

GET: <https://{endpoint}/v3/0605767b5780d5762fc5c0118072a564/checkpoints/d9ce6924-d753-4132-bd16-a9f8838ea7d2>

- 响应示例

```
{  
    "checkpoint": {  
        "id": "d9ce6924-d753-4132-bd16-a9f8838ea7d2",  
        "project_id": "0605767b5780d5762fc5c0118072a564",  
        "status": "available",  
        "vault": null,  
        "created_at": "2020-08-17T06:49:06.260790",  
    }  
}
```

```
        "extra_info": null
    }
```

5.2 示例 2：创建存储库自动备份

场景描述

本章节指导用户通过策略接口，设置备份策略，绑定策略到存储库实现自动备份。

涉及接口

- **创建策略**：创建执行备份的周期。
- **设置存储库策略**：关联待设置的策略到存储库。

操作步骤

1. 创建策略。

- 接口相关信息

URI格式：POST /v3/{project_id}/policies

详情请参见：[创建策略](#)。

- 请求示例

POST: <https://{{endpoint}}/v3/0605767b5780d5762fc5c0118072a564/policies>

{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{
    "policy": {
        "name": "dh_test_policy",
        "trigger": {
            "properties": {
                "pattern": [
                    "FREQ=WEEKLY;BYDAY=SU,MO,TU,WE,TH,FR,SA;BYHOUR=23;BYMINUTE=00"
                ]
            },
            "operation_definition": {
                "retention_duration_days": 30
            }
        }
    }
}
```

- 响应示例

```
{
    "policy": {
        "id": "30d7cf2d-14fc-415b-b7da-858b37f47250",
        "name": "dh_test_policy",
        "operation_type": "backup",
        "operation_definition": {
            "retention_duration_days": 30
        },
        "enabled": true,
        "trigger": {
            "id": "7954175b-ef2c-432c-b936-f6c83df7a593",
            "name": "default",
            "type": "time",
            "properties": {
                "pattern": [
                    "FREQ=WEEKLY;BYDAY=SU,MO,TU,WE,TH,FR,SA;BYHOUR=23;BYMINUTE=00"
                ]
            }
        }
    }
}
```

```
        ],
        "start_time": "2020-08-17 08:39:44"
    },
},
"associated_vaults": null
}
```

2. 设置存储库策略。

- 接口相关信息

POST /v3/{project_id}/vaults/{vault_id}/associatepolicy

详情请参见[设置存储库策略](#)。

- 请求示例

POST: <https://{{endpoint}}/v3/0605767b5780d5762fc5c0118072a564/vaults/ea7b8717-2543-478a-a92d-3ca7ee448f67/associatepolicy>

{endpoint}信息请从[地区和终端节点](#)获取。

Body:

```
{
    "policy_id": "30d7cf2d-14fc-415b-b7da-858b37f47250"
}
```

- 响应示例

```
{
    "associate_policy": {
        "vault_id": "ea7b8717-2543-478a-a92d-3ca7ee448f67",
        "policy_id": "30d7cf2d-14fc-415b-b7da-858b37f47250"
    }
}
```

5.3 示例 3：查询备份信息

场景描述

本章节指导用户查询所有备份接口，使用分页参数查询租户所有的备份。

本节示例操作包括分页查询，接口支持过滤和排序参数，查询指定资源备份的操作，参数详情请参见[查询所有备份](#)。

涉及接口

本示例场景涉及如下接口调用：

- [指定的limit和offset查询所有备份列表](#)
- [指定的资源类型查询所有备份列表](#)

操作步骤

1. 指定的limit和offset查询所有备份列表。

- 接口相关信息

URI格式: GET /v3/{project_id}/backups

详情请参见[查询所有备份](#)。

- 请求示例

GET:<https://{{endpoint}}/v3/0605767b5780d5762fc5c0118072a564/backups?limit=100&offset=0>

{endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{  
    "backups": [  
        .....  
        {  
            "children": [],  
            "checkpoint_id": "e6aec7a9-7b03-4c1d-8a07-5983b53c53f3",  
            "created_at": "2020-08-18T06:00:45.375070",  
            "description": null,  
            "expired_at": null,  
            "extend_info": {  
                "app_consistency": {  
                    "app_consistency": "0",  
                    "app_consistency_status": "0",  
                    "app_consistency_error_code": "0",  
                    "app_consistency_error_message": ""  
                },  
                "auto_trigger": true,  
                "bootable": null,  
                "os_images_data": null,  
                "progress": null,  
                "snapshot_id": null,  
                "support_lld": false,  
                "supported_restore_mode": "backup",  
                "system_disk": false,  
                "contain_system_disk": true,  
                "architecture": "x86_64"  
            },  
            "id": "62617971-839d-4d23-8dfd-4ca65c039bdf",  
            "image_type": "backup",  
            "name": "autobk_cf91_0003",  
            "parent_id": null,  
            "project_id": "0605767b5780d5762fc5c0118072a564",  
            "protected_at": "2020-08-18T06:01:10.432117",  
            "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
            "resource_az": "br-iaas-odin1a",  
            "resource_id": "d6bf7592-ca52-43a2-9979-e418d64b29bb",  
            "resource_name": "xzl_ecs-0003-0001",  
            "resource_size": 40,  
            "resource_type": "OS::Nova::Server",  
            "status": "available",  
            "updated_at": "2020-08-18T06:06:44.928325",  
            "vault_id": "1572bd27-e221-4f28-94ca-9777d232fc7",  
            "replication_records": []  
        }  
    ],  
    "count": 1663  
}
```

2. 指定的资源类型查询所有备份列表。

- 接口相关信息

URI格式: GET /v3/{project_id}/backups

接口与[步骤1](#)保持一致。

- 请求示例

GET: [https://\[endpoint\]/v3/0605767b5780d5762fc5c0118072a564/backups?resource_type=OS::Nova::Server&limit=5&offset=0](https://[endpoint]/v3/0605767b5780d5762fc5c0118072a564/backups?resource_type=OS::Nova::Server&limit=5&offset=0)

{endpoint}信息请从[地区和终端节点](#)获取。

- 响应示例

```
{  
    "backups": [  
        .....  
        {  
            "children": [],  
            "checkpoint_id": "e6aec7a9-7b03-4c1d-8a07-5983b53c53f3",  
            "created_at": "2020-08-18T06:00:45.375070",  
            "description": null,  
            "expired_at": null,  
            "extend_info": {  
                "app_consistency": {  
                    "app_consistency": "0",  
                    "app_consistency_status": "0",  
                    "app_consistency_error_code": "0",  
                    "app_consistency_error_message": ""  
                },  
                "auto_trigger": true,  
                "bootable": null,  
                "os_images_data": null,  
                "progress": null,  
                "snapshot_id": null,  
                "support_lld": false,  
                "supported_restore_mode": "backup",  
                "system_disk": false,  
                "contain_system_disk": true,  
                "architecture": "x86_64"  
            },  
            "id": "62617971-839d-4d23-8dfd-4ca65c039bdf",  
            "image_type": "backup",  
            "name": "autobk_cf91_0003",  
            "parent_id": null,  
            "project_id": "0605767b5780d5762fc5c0118072a564",  
            "protected_at": "2020-08-18T06:01:10.432117",  
            "provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",  
            "resource_az": "br-iaas-odin1a",  
            "resource_id": "d6bf7592-ca52-43a2-9979-e418d64b29bb",  
            "resource_name": "xzl_ecs-0003-0001",  
            "resource_size": 40,  
            "resource_type": "OS::Nova::Server",  
            "status": "available",  
            "updated_at": "2020-08-18T06:06:44.928325",  
            "vault_id": "1572bd27-e221-4f28-94ca-9777d232fc7",  
            "replication_records": []  
        }  
    ],  
    "count": 1663  
}
```

```
"checkpoint_id": "e328d05e-4b28-4898-b8c1-2bfe6621ec03",
"created_at": "2020-08-18T07:00:46.932061",
"description": null,
"expired_at": null,
"extend_info": {
    "app_consistency": {
        "app_consistency": "0",
        "app_consistency_status": "0",
        "app_consistency_error_code": "0",
        "app_consistency_error_message": ""
    },
    "auto_trigger": true,
    "bootable": null,
    "os_images_data": null,
    "progress": null,
    "snapshot_id": null,
    "support_lld": false,
    "supported_restore_mode": "backup",
    "system_disk": false,
    "contain_system_disk": true,
    "architecture": "x86_64"
},
"id": "c892ed58-3a18-47c2-9e31-a1d543dc490a",
"image_type": "backup",
"name": "autobk_7234_0003",
"parent_id": null,
"project_id": "0605767b5780d5762fc5c0118072a564",
"protected_at": "2020-08-18T07:01:12.675112",
"provider_id": "0daac4c5-6707-4851-97ba-169e36266b66",
"resource_az": "br-iaas-odin1a",
"resource_id": "d6bf7592-ca52-43a2-9979-e418d64b29bb",
"resource_name": "xzl_ecs-0003-0001",
"resource_size": 40,
"resource_type": "OS::Nova::Server",
"status": "available",
"updated_at": "2020-08-18T07:06:47.518054",
"vault_id": "1572bd27-e221-4f28-94ca-9777d232fc7",
"replication_records": []
}
],
"count": 150
}
```

6 权限和授权项

6.1 权限及授权项说明

如果您需要对您所拥有的CBR进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为云帐号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用CBR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。

帐号具备所有接口的调用权限，如果使用帐号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询云服务器列表，那么这个IAM用户被授予的策略中必须包含允许“ecs:servers:list”的授权项，该接口才能调用成功。

支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。

- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。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，“×”表示暂不支持。

CBR的支持自定义策略授权项如下所示：

- 【示例】**存储库**，包含CBR所有存储库接口对应的授权项，如创建存储库、查询存储库列表、修改存储库、删除存储库、添加资源、移除资源等接口。
- 【示例】**备份共享**，包括CBR备份共享接口对应的授权项，如添加备份成员、获取备份成员列表、更新备份成员状态等接口。

6.2 CBR 授权项分类

任务

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询任务 列表	GET /v3/ {project_id}/ operation-logs	cbr:tasks:list	√	√
查询单个 任务	GET /v3/ {project_id}/ operation-logs/ {operation_log_ id}	cbr:tasks:ge t	√	√

可保护性

权限	对应API接口	授权项	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询可保护资源	GET /v3/{project_id}/protectables/{protectable_type}/instances	cbr:vaults:listProtectables	ecs:cloudServers:list evs:volumes:list	√	√
查询指定可保护资源	GET /v3/{project_id}/protectables/{protectable_type}/instances/{instance_id}	cbr:vaults:getProtectables	ecs:cloudServers:list evs:volumes:list	√	√
查询复制能力	GET /v3/{project_id}/replication-capabilities	cbr:backups:queryReplicationCapability	-	√	√

存储库

权限	对应API接口	授权项	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
设置存储库策略	POST /v3/{project_id}/vaults/{vault_id}/associatepolicy	cbr:vaults:associatePolicy	-	√	√
查询指定存储库	GET /v3/{project_id}/vaults/{vault_id}	cbr:vaults:get	-	√	√
修改存储库	PUT /v3/{project_id}/vaults/{vault_id}	cbr:vaults:update	-	√	√

权限	对应API接口	授权项	依赖的授权项	IAM项目(Project)	企业项目(Enterprise Project)
删除存储库	DELETE /v3/{project_id}/vaults/{vault_id}	cbr:vaults:delete	-	√	√
移除资源	POST /v3/{project_id}/vaults/{vault_id}/removeResources	cbr:vaults:removeResources	-	√	√
添加资源	POST /v3/{project_id}/vaults/{vault_id}/addresources	cbr:vaults:addResources	ecs:cloudServers:list evs:volumes:list	√	√
查询存储库列表	GET /v3/{project_id}/vaults	cbr:vaults:list	-	√	√
创建存储库	POST /v3/{project_id}/vaults	cbr:vaults:create	ecs:cloudServers:list evs:volumes:list	√	√
解除存储库策略	POST /v3/{project_id}/vaults/{vault_id}/dissociatePolicy	cbr:vaults:dissociatePolicy	-	√	√

还原点

权限	对应API接口	授权项	依赖的授权项	IAM项目(Project)	企业项目(Enterprise Project)
同步备份还原点	POST /v3/{project_id}/checkpoints-sync	cbr:vaults:sync	-	√	√

权限	对应API接口	授权项	依赖的授权项	IAM项目(Project)	企业项目(Enterprise Project)
复制备份还原点	POST /v3/{project_id}/checkpoints/replicate	cbr:vaults:replicate	-	√	√
创建备份还原点	POST /v3/{project_id}/checkpoints	cbr:vaults:backup	ecs:cloudServers:list evs:volumes:list	√	√

备份共享

权限	对应API接口	授权项	IAM项目(Project)	企业项目(Enterprise Project)
添加备份成员	POST /v3/{project_id}/backups/{backup_id}/members	cbr:member:create	√	√
更新备份成员状态	PUT /v3/{project_id}/backups/{backup_id}/members/{member_id}	cbr:member:update	√	√
获取备份成员详情	GET /v3/{project_id}/backups/{backup_id}/members/{member_id}	cbr:member:get	√	√
获取备份成员列表	GET /v3/{project_id}/backups/{backup_id}/members	cbr:member:list	√	√

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
删除指定的备份成员	DELETE /v3/{project_id}/backups/{backup_id}/members/{member_id}	cbr:member:delete	√	√

备份

权限	对应API接口	授权项	依赖的授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询所有备份	GET /v3/{project_id}/backups	cbr:backups:list	-	√	√
查询指定备份	GET /v3/{project_id}/backups/{backup_id}	cbr:backups:get	-	√	√
删除备份	DELETE /v3/{project_id}/backups/{backup_id}	cbr:backups:delete	-	√	√
同步备份	POST /v3/{project_id}/backups/sync	cbr:backups:sync	-	√	√
备份恢复	POST /v3/{project_id}/backups/{backup_id}/restore	cbr:backups:restore	ecs:cloudServers:list evs:volumes:list	√	√
复制备份	POST /v3/{project_id}/backups/{backup_id}/replicate	cbr:backups:replicate	-	√	√

策略

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询策略列表	GET /v3/{project_id}/policies	cbr:policies:list	√	√
创建策略	POST /v3/{project_id}/policies	cbr:policies:create	√	√
查询单个策略	GET /v3/{project_id}/policies/{policy_id}	cbr:policies:get	√	√
修改策略	PUT /v3/{project_id}/policies/{policy_id}	cbr:policies:update	√	√
删除策略	DELETE /v3/{project_id}/policies/{policy_id}	cbr:policies:delete	√	√

标签

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
查询存储库资源实例	POST /v3/{project_id}/vault/resource_instances/action	cbr:vaults:listResources	√	√
批量添加删除存储库资源标签	POST /v3/{project_id}/vault/{vault_id}/tags/action	cbr:vaults:bulkCreateOrDeleteTags	√	√
添加存储库资源标签	POST /v3/{project_id}/vault/{vault_id}/tags	cbr:vaults:setTags	√	√

权限	对应API接口	授权项	IAM项目 (Project)	企业项目 (Enterprise Project)
删除存储库资源标签	DELETE /v3/{project_id}/vault/{vault_id}/tags/{key}	cbr:vaults:deleteTags	√	√
查询存储库资源标签	GET /v3/{project_id}/vault/{vault_id}/tags	cbr:vaults:getTags	√	√
查询存储库项目标签	GET /v3/{project_id}/vault/tags	cbr:vaults:listProjectTags	√	√

7 附录

7.1 状态码

- 正常

状态码	说明
200 OK	GET和PUT操作正常返回。
201 Created	POST操作正常返回。
202 Accepted	请求已被接受。
204 No Content	DELETE操作正常返回。

- 异常

状态码	说明
400 Bad Request	服务器未能处理请求。
401 Unauthorized	被请求的页面需要用户名和密码。
403 Forbidden	对被请求页面的访问被禁止。
404 Not Found	服务器无法找到被请求的页面。
405 Method Not Allowed	请求中指定的方法不被允许。
406 Not Acceptable	服务器生成的响应无法被客户端所接受。
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。
500 Internal Server Error	请求未完成。服务异常。

状态码	说明
501 Not Implemented	请求未完成。服务器不支持所请求的功能。
502 Bad Gateway	请求未完成。服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成。系统暂时异常。
504 Gateway Timeout	网关超时。

7.2 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.0001	没有可以备份的资源	没有可以备份的资源	确认资源是否正在备份或联系技术支持
400	BackupService.e.1011	目标项目不支持复制	目标项目不支持复制	请联系技术支持。
400	BackupService.e.1012	复制已经达到上限，不需要在复制	复制已经达到上限，不需要在复制	请检查资源复制项的数量是否已达到上限。
400	BackupService.e.2001	云服务器挂载的云硬盘的容量小于备份盘，不能执行恢复操作。	云服务器挂载的云硬盘的容量小于备份盘，不能执行恢复操作。	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.2002	待恢复的资源正在备份中，请等待备份结束后再进行操作。	待恢复的资源正在备份中，请等待备份结束后再进行操作。	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.2003	不允许备份恢复到其他云服务器。	不允许备份恢复到其他云服务器。	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.2004	复制的备份不支持恢复。	复制的备份不支持恢复。	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.2005	备份状态不允许恢复。	备份状态不允许恢复。	请明确具体操作约束，按照约束进行操作。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.2006	ECS备份不允许恢复到BMS。	ECS备份不允许恢复到BMS。	请明确具体操作约束，按照约束进行操作。
400	BackupService.2007	Terminated状态ECS不允许恢复。	Terminated状态ECS不允许恢复。	请先确认ECS是否可用，再进行备份。
400	BackupService.2008	ECS状态不支持恢复。	ECS状态不支持恢复。	请确认ECS的状态。
400	BackupService.2009	卷类型不支持恢复。	卷类型不支持恢复。	请明确具体操作约束，按照约束进行操作。
400	BackupService.2010	资源正在恢复中。	资源正在恢复中。	请稍后再试。
400	BackupService.2011	卷状态不支持恢复。	卷状态不支持恢复。	请确认卷状态。
400	BackupService.2012	不支持部分磁盘备份恢复。	不支持部分磁盘备份恢复。	请联系技术支持。
400	BackupService.2013	不支持将数据盘备份恢复到系统盘	不支持将数据盘备份恢复到系统盘	请明确具体操作限制，使用合法的恢复参数。
400	BackupService.2014	BMS备份不能恢复到ECS	BMS备份不能恢复到ECS	请选择正确的备份进行恢复。
400	BackupService.2015	架构信息不同不支持恢复。	架构信息不同不支持恢复。	请选择正确架构进行恢复。
400	BackupService.4002	迁移失败。	迁移失败。	请稍后再试。
400	BackupService.4003	迁移过的租户不允许使用CSBS/VBS服务。	迁移过的租户不允许使用CSBS/VBS服务。	请使用新服务。
400	BackupService.6001	该类型的策略数量达到上限	该类型的策略数量达到上限	请联系技术支持。
400	BackupService.6003	策略已绑定复制存储库，不能修改复制目标区域	策略已绑定复制存储库，不能修改复制目标区域	请将复制存储库解绑后重试。
400	BackupService.6100	存储库资源数量超过限制	存储库资源数量超过限制	合理创建存储库，将未绑定的资源绑定到新的存储库。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.6101	存储库容量非法	存储库容量非法	请明确具体操作限制，使用合法的存储库参数。
400	BackupService.e.6102	存储库不支持该资源类型。	存储库不支持该资源类型。	请明确具体操作限制，使用合法的存储库类型。
400	BackupService.e.6103	资源已绑定存储库	资源已绑定存储库	请明确具体操作限制，使用合法的资源参数。
400	BackupService.e.6104	存储库资源重复。	存储库资源重复。	请明确具体操作限制，使用合法的资源参数。
400	BackupService.e.6106	存储库重复。	存储库重复。	请明确具体操作限制，使用合法的存储库参数。
400	BackupService.e.6107	存储库扩容失败。	存储库扩容失败。	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6108	存储库无法添加资源。	存储库无法添加资源。	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6109	账单不存在。	账单不存在。	请联系技术支持。
400	BackupService.e.6110	存储库不能更新。	存储库不能更新。	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6111	存储库不能删除。	存储库不能删除。	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6112	存储库状态非法。	存储库状态非法。	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.e.6113	存储库状态不支持备份。	存储库状态不支持备份。	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.e.6114	存储库已使用容量超过最大容量。	存储库已使用容量超过最大容量。	请合法使用存储库。
400	BackupService.e.6115	删除存储库备份失败	删除存储库备份失败	请重试，若仍未解决，请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.6116	不支持的保护类型	不支持的保护类型	请使用合法的protect type。
400	BackupService.e.6117	未知的策略类型	未知的策略类型	请使用合法的OperationType。
400	BackupService.e.6118	检查目标端存储库失败	检查目标端存储库失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6119	目标端存储库不支持复制	目标端存储库不支持复制	请确认目标端存储库是否支持复制。
400	BackupService.e.6120	目标端存储库不存在	目标端存储库不存在	请确认目标端存储库是否存在。
400	BackupService.e.6121	存储库删除失败	存储库删除失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6122	存储库的资源类型不支持复制	存储库的资源类型不支持复制	请确认资源类型能进行复制操作。
400	BackupService.e.6123	扩容失败，可能存在未处理的扩容订单或者扩容正在处理	扩容失败，可能存在未处理的扩容订单或者扩容正在处理	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6124	存储库类型不支持备份	存储库类型不支持备份	请合理使用不同类型的存储库。
400	BackupService.e.6125	存储库正在执行备份	存储库正在执行备份	请等待上一次备份任务完成后再次执行备份操作。
400	BackupService.e.6126	该存储库不支持数据库备份	该存储库不支持数据库备份	请合理使用不同类型的存储库。
400	BackupService.e.6127	该存储库不支持绑定该策略	该存储库不支持绑定该策略	请合理使用不同类型的存储库。
400	BackupService.e.6128	该存储库的类型不支持复制	该存储库的类型不支持复制	请使用合理的的存储库进行复制。
400	BackupService.e.6129	目标存储库的容量到达上限	目标存储库的容量到达上限	请先扩容目标存储库的容量，再进行此操作
400	BackupService.e.6130	存储库正在复制中	存储库正在复制中	请稍后重试。
400	BackupService.e.6131	流量记录不存在	流量记录不存在	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.6133	存储库数量超过限制	存储库数量超过限制	具体原因请联系技术支持。
400	BackupService.e.6134	存储库中的资源正在移除中	存储库中的资源正在移除中	请稍后重试。
400	BackupService.e.6135	资源不存在于存储库中	资源不存在于存储库中	请确认该资源是否绑定到该存储库。
400	BackupService.e.6136	混合云备份存储库不支持绑定备份策略	混合云备份存储库不支持绑定备份策略	请联系技术支持。
400	BackupService.e.6140	加密卷不能指定为系统盘	加密卷不能指定为系统盘	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.6141	Scsi卷不能指定为系统盘	Scsi卷不能指定为系统盘	请明确具体操作约束，按照约束进行操作。
400	BackupService.e.6142	所有存储库容量已超过上限	所有存储库容量已超过上限	请扩容后再执行操作。
400	BackupService.e.6201	备份不能删除	备份不能删除	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6202	存储库状态非可用时备份不能用于恢复	存储库状态非可用时备份不能用于恢复	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.e.6203	正在执行备份	正在执行备份	请在备份任务完成后再次执行备份。
400	BackupService.e.6204	备份已注册镜像不能删除	备份已注册镜像不能删除	请先删除对应的镜像后再删除备份。
400	BackupService.e.6205	资源在老的服务已存在备份。	资源在老的服务已存在备份。	具体原因请联系技术支持。
400	BackupService.e.6206	备份在当前状态下不支持元数据查询。	备份在当前状态下不支持元数据查询。	请稍后再试。
400	BackupService.e.6215	资源状态不支持备份	资源状态不支持备份	请确认资源状态是否支持备份
400	BackupService.e.6216	备份正在使用，不能删除	备份正在使用，不能删除	请稍后再试。
400	BackupService.e.6300	资源类型与备份提供商不匹配	资源类型与备份提供商不匹配	具体原因请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.6301	备份提供商ID非法	备份提供商ID非法	请使用合法的 provider_id。
400	BackupService.e.6400	创桶失败	创桶失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6401	设置桶配额失败	设置桶配额失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6403	获取桶的使用量失败	获取桶的使用量失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6404	删除桶失败	删除桶失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6405	删除桶中的对象失败	删除桶中的对象失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6406	列举桶中的对象失败	列举桶中的对象失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6407	设置桶的ACL失败	设置桶的ACL失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6408	设置桶的策略失败	设置桶的策略失败	请重试，若仍未解决，请联系技术支持。
400	BackupService.e.6600	资源的标签数量超过限制	资源的标签数量超过限制	请删除该资源的部分标签后重试。
400	BackupService.e.6700	仅支持云服务器备份共享。	仅支持云服务器备份共享。	请使用云服务器备份进行共享。
400	BackupService.e.6701	备份共享个数超过限制。	备份共享个数超过限制。	请确认该备份共享个数是否超过限制。
400	BackupService.e.6702	非可用状态的备份不支持共享。	非可用状态的备份不支持共享。	请使用可用备份进行共享。
400	BackupService.e.6703	备份重复共享	备份重复共享	请确认该备份是否已共享给该成员。
400	BackupService.e.6704	共享租户项目id非法	共享租户项目id非法	请使用正确的项目id进行共享。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.e.6706	备份成员状态更新参数非法。	备份成员状态更新参数非法。	请确认更新参数是否正确。
400	BackupService.e.6707	不允许备份共享	不允许备份共享	请确认是否支持备份共享。
400	BackupService.e.6708	更新备份成员状态失败。	更新备份成员状态失败。	请联系技术支持。
400	BackupService.e.6709	含有加密卷备份的云服务器备份不支持共享	含有加密卷备份的云服务器备份不支持共享	请明确具体操作约束, 按照约束进行操作。
400	BackupService.e.6710	该共享备份已经注册镜像, 请先删除镜像。	该共享备份已经注册镜像, 请先删除镜像。	该共享备份已经注册镜像, 请先删除镜像。
400	BackupService.e.6711	删除共享备份失败。	删除共享备份失败。	请联系技术支持。
400	BackupService.e.6712	非法的存储库状态不支持备份共享。	非法的存储库状态不支持备份共享。	请明确具体操作约束, 按照约束进行操作。
400	BackupService.e.6713	接受共享备份的存储库类型与备份的资源类型不同。	接受共享备份的存储库类型与备份的资源类型不同。	请使用相同类型的存储库和备份资源。
400	BackupService.e.6714	共享备份不支持再次共享。	共享备份不支持再次共享。	请明确具体操作约束, 按照约束进行操作。
400	BackupService.e.7001	DESS磁盘不支持备份	DESS磁盘不支持备份	请明确具体操作约束, 按照约束进行操作。
400	BackupService.e.7002	SCSI磁盘不支持备份	SCSI磁盘不支持备份	请明确具体操作约束, 按照约束进行操作。
400	BackupService.e.7003	磁盘状态不支持备份	磁盘状态不支持备份	请在正确的状态下执行该操作, 明确具体操作约束。
400	BackupService.e.7004	该磁盘不支持备份或恢复	该磁盘不支持备份或恢复	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.7005	云硬盘已经加入老云硬盘备份服务中。	云硬盘已经加入老云硬盘备份服务中。	请从老云硬盘备份服务卸载磁盘，再挂载到该存储库上。
400	BackupService.7006	磁盘已存在于存储库中	磁盘已存在于存储库中	请从存储库中卸载磁盘，在进行此操作。
400	BackupService.7007	很早之前创建的卷不支持备份。	很早之前创建的卷不支持备份。	请更换卷后执行备份。
400	BackupService.7008	容灾端的卷不支持恢复操作。	容灾端的卷不支持恢复操作。	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.7101	服务器状态不支持备份	服务器状态不支持备份	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.7102	服务器已终止	服务器已终止	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.7103	该服务器不支持备份	该服务器不支持备份	具体原因请联系技术支持。
400	BackupService.7104	服务器包含SCSI磁盘不支持备份	服务器包含SCSI磁盘不支持备份	请移除SCSI类型的磁盘重新备份。
400	BackupService.7105	磁盘后端不一致	磁盘后端不一致	请联系技术支持。
400	BackupService.7106	共享磁盘不支持备份	共享磁盘不支持备份	请明确具体操作约束，按照约束进行操作。
400	BackupService.7107	共享磁盘数量超过限制	共享磁盘数量超过限制	请排除共享类型的磁盘重新备份。
400	BackupService.7108	服务器没有挂载磁盘不支持备份	服务器没有挂载磁盘不支持备份	请先挂载磁盘到服务器再进行备份。
400	BackupService.7109	不支持裸金属服务器备份	不支持裸金属服务器备份	请明确具体操作约束，按照约束进行操作。
400	BackupService.7110	资源类型与provider_id不匹配	资源类型与provider_id不匹配	请使用合法的provider_id。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.7111	服务器包含DESS磁盘不支持备份	服务器包含DESS磁盘不支持备份	请移除DESS类型的磁盘重新备份。
400	BackupService.7112	该服务器已经绑定到老云服务器备份服务中	该服务器已经绑定到老云服务器备份服务中	请从老云服务器备份服务中移除云服务器，再挂载到该存储库上。
400	BackupService.7113	不支持裸金属服务器本地盘备份	不支持裸金属服务器本地盘备份	请排除本地盘后重新备份。
400	BackupService.7114	只允许恢复到同一个操作系统的云服务器。	只允许恢复到同一个操作系统的云服务器。	请明确具体操作限制，使用合法的恢复参数。
400	BackupService.7115	备份的服务器与恢复的目标服务器类型不同	备份的服务器与恢复的目标服务器类型不同	请明确具体操作限制，使用合法的恢复参数。
400	BackupService.7116	服务器已存在于存储库中	服务器已存在于存储库中	请从存储库中卸载云服务器，再进行此操作。
400	BackupService.7117	容灾端服务器不允许做恢复操作。	容灾端服务器不允许做恢复操作。	请明确具体操作限制，使用合法的恢复参数。
400	BackupService.7200	云数据库备份中。	云数据库备份中。	请稍后再试。
400	BackupService.7201	磁盘未挂载到服务器。	磁盘未挂载到服务器。	请确认后重试。
400	BackupService.7203	快照的状态不可用。	快照的状态不可用。	请稍后再试。
400	BackupService.7204	快照与磁盘不匹配。	快照与磁盘不匹配。	请确认后重试。
400	BackupService.7300	保管库不支持同步	保管库不支持同步	请确认存储库类型是否支持同步。
400	BackupService.7301	存储库不为空	存储库不为空	请明确具体操作约束，按照约束进行操作。
400	BackupService.7302	同步备份中	同步备份中	请明确具体操作约束，按照约束进行操作。

状态码	错误码	错误信息	描述	处理措施
400	BackupService.7303	存在库数量超出限制。	存在库数量超出限制。	请联系技术支持
400	BackupService.7501	弹性文件系统正在备份中	弹性文件系统正在备份中	请稍后重试
400	BackupService.7502	弹性文件系统状态不支持备份	弹性文件系统状态不支持备份	请在正确的状态下执行该操作，明确具体操作约束
400	BackupService.7503	弹性文件系统子状态不支持备份	弹性文件系统子状态不支持备份	请在正确的状态下执行该操作，明确具体操作约束
400	BackupService.7504	弹性文件系统冻结失败	弹性文件系统冻结失败	稍后请重试
400	BackupService.7505	弹性文件系统解冻失败	弹性文件系统解冻失败	稍后请重试
400	BackupService.7506	弹性文件系统执行任务失败	弹性文件系统执行任务失败	稍后请重试，或联系技术支持
400	BackupService.7507	弹性文件系统与备份时不一致无法恢复	弹性文件系统与备份时不一致无法恢复	请在正确的状态下执行该操作，明确具体操作约束
400	BackupService.7508	弹性文件系统预恢复失败	弹性文件系统预恢复失败	请重试，若仍未解决，请联系技术支持
400	BackupService.7509	弹性文件系统延迟恢复失败	弹性文件系统延迟恢复失败	请重试，若仍未解决，请联系技术支持
400	BackupService.7510	弹性文件系统备份只能恢复到原弹性文件系统。	弹性文件系统备份只能恢复到原弹性文件系统。	请恢复到原弹性文件系统。
400	BackupService.7511	弹性文件系统状态不支持恢复。	弹性文件系统状态不支持恢复。	请在正确的状态下执行该操作，明确具体操作约束。
400	BackupService.8300	快照配额不足	快照配额不足	请扩容后重试
400	BackupService.9900	参数校验失败	参数校验失败	请使用合法的参数。
403	BackupService.8600	未实名认证	未实名认证	请先进行实名认证。
404	BackupService.4001	迁移记录不存在	迁移记录不存在	请提供正确的迁移记录id

状态码	错误码	错误信息	描述	处理措施
404	BackupService.6000	策略不存在	策略不存在	请确认策略是否存在
404	BackupService.6002	存储库和策略的绑定关系不存在	存储库和策略的绑定关系不存在	请先为存储库设置备份策略
404	BackupService.6105	存储库不存在	存储库不存在	请明确具体操作限制，使用合法的存储库参数
404	BackupService.6200	该备份不存在	该备份不存在	请确认该备份是否存在。
404	BackupService.6217	备份还原点不存在	备份还原点不存在	请确认还原点是否存在。
404	BackupService.6302	资源不存在。	资源不存在。	请确认查询内容。
404	BackupService.6402	桶不为空	桶不为空	请清空桶（删除备份和备份计划）
404	BackupService.6500	操作日志不存在	操作日志不存在	请确认该操作任务是否存在。
404	BackupService.6501	任务不存在	任务不存在	请确认任务是否存在。
404	BackupService.6601	键不存在	键不存在	请输入正确的键名。
404	BackupService.6705	备份成员不存在	备份成员不存在	请确认备份共享成员是否存在。
404	BackupService.7000	卷不存在	卷不存在	请明确具体操作约束，按照约束进行操作。
404	BackupService.7100	服务器不存在	服务器不存在	请在正确的状态下执行该操作，明确具体操作约束。
404	BackupService.7202	快照获取失败	快照获取失败	请确认查询参数
404	BackupService.7500	弹性文件系统不存在	弹性文件系统不存在	请确认文件系统是否存在
500	BackupService.0002	资源正在备份	资源正在备份	请稍后重试。
500	BackupService.1001	当前备份不是可用状态，不支持复制	当前备份不是可用状态，不支持复制	请检查备份状态是否支持复制

状态码	错误码	错误信息	描述	处理措施
500	BackupService.e.1002	当前备份的类型不支持复制。	当前备份的类型不支持复制。	请确认镜像类型是backup或sync。
500	BackupService.e.1003	当前备份的服务器不是弹性云服务器，不支持复制	当前备份的服务器不是弹性云服务器，不支持复制	请确认当前备份服务器是弹性云服务器
500	BackupService.e.1004	当前备份的弹性云服务器不包含系统盘，不支持复制	当前备份的弹性云服务器不包含系统盘，不支持复制	请确认当前服务器包含系统盘
500	BackupService.e.1005	目标区域不支持复制	目标区域不支持复制	请确实当前区域是否支持复制
500	BackupService.e.1006	导入复制失败	导入复制失败	联系技术支持
500	BackupService.e.1007	无法检测当前备份是否已复制到目标区域，不支持复制	无法检测当前备份是否已复制到目标区域，不支持复制	请稍后重试
500	BackupService.e.1008	无法检测目标区域的存储库，不支持复制	无法检测目标区域的存储库，不支持复制	请稍后重试。
500	BackupService.e.1009	当前备份正在复制或已经复制到目标区域	当前备份正在复制或已经复制到目标区域	请确认目标区域复制是否已存在。
500	BackupService.e.1013	该备份副本系统盘由用户指定，暂不支持复制	该备份副本系统盘由用户指定，暂不支持复制	请选择其他备份进行复制
500	BackupService.e.4004	清理数据失败	清理数据失败	请稍后再试。
500	BackupService.e.4005	检查目标存储库失败	检查目标存储库失败	请稍后再试。
500	BackupService.e.4006	迁移非稳态副本失败	迁移非稳态副本失败	请稍后再试。
500	BackupService.e.4007	检查其他region的迁移进度失败	检查其他region的迁移进度失败	请稍后再试。

状态码	错误码	错误信息	描述	处理措施
500	BackupService.e.6132	创建存储库失败	创建存储库失败	请重试，若仍未解决，请联系技术支持。
500	BackupService.e.6137	云服务器的卷不能在多个存储库中备份	云服务器的卷不能在多个存储库中备份	请不要选择云服务器下相同的卷绑定到不同存储库
500	BackupService.e.6138	从CBC获取语言首选项失败	从CBC获取语言首选项失败	仅支持中英文。
500	BackupService.e.6139	获取xdomain_type失败	获取xdomain_type失败	请稍后重试
500	BackupService.e.6207	该类型的备份不支持创建镜像	该类型的备份不支持创建镜像	请明确具体操作约束，按照约束进行操作。
500	BackupService.e.6208	备份在当前状态下不支持创建镜像	备份在当前状态下不支持创建镜像	请等待一段时间后重试或联系技术支持
500	BackupService.e.6209	该备份中不存在系统盘数据，无法创建镜像	该备份中不存在系统盘数据，无法创建镜像	请明确具体操作约束，按照约束进行操作
500	BackupService.e.6210	当前备份已创建过镜像，无法重复创建	当前备份已创建过镜像，无法重复创建	请明确具体操作约束，按照约束进行操作
500	BackupService.e.6211	当前备份已创建过镜像，无法重复创建	当前备份已创建过镜像，无法重复创建	请明确具体操作约束，按照约束进行操作
500	BackupService.e.6212	镜像创建失败	镜像创建失败	请联系技术支持
500	BackupService.e.6213	备份未注册到指定的镜像	备份未注册到指定的镜像	请联系技术支持
500	BackupService.e.6214	镜像解除注册失败	镜像解除注册失败	请联系技术支持。
500	BackupService.e.7009	卷备份缓慢加载中	卷备份缓慢加载中	请在正确的状态下执行该操作，明确具体操作约束
500	BackupService.e.8400	从cbc获取产品失败	从cbc获取产品失败	请稍后再试
500	BackupService.e.9910	未知异常	未知异常	具体原因请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
500	BackupService.9998	鉴权失败	鉴权失败	确认用户信息。

7.3 获取项目 ID

操作场景

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。有如下两种获取方式：

- [调用API获取项目ID](#)
- [从控制台获取项目ID](#)

调用 API 获取项目 ID

项目ID可以通过调用[查询指定条件下的项目列表](#)API获取。

获取项目ID的接口为“GET `https://{{Endpoint}}/v3/projects`”，其中{{Endpoint}}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目ID。

```
{  
    "projects": [  
        {  
            "domain_id": "65382450e8f64ac0870cd180d14e684b",  
            "is_domain": false,  
            "parent_id": "65382450e8f64ac0870cd180d14e684b",  
            "name": "project_name",  
            "description": "",  
            "links": {  
                "next": null,  
                "previous": null,  
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"  
            },  
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",  
            "enabled": true  
        }  
    ],  
    "links": {  
        "next": null,  
        "previous": null,  
        "self": "https://www.example.com/v3/projects"  
    }  
}
```

从控制台获取项目 ID

从控制台获取项目ID的步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。
在“API凭证”页面的项目列表中查看项目ID。

图 7-1 查看项目 ID

The screenshot shows the 'API凭证' (API Credential) page. On the left, there's a sidebar with '我的凭证' (My Credentials) and '访问密钥' (Access Keys). The main area has tabs for 'API凭证' (selected) and '访问密钥'. It displays IAM user information: 'IAM用户名' (User Name) and '账号名' (Account Name), and 'IAM用户ID' (User ID) and '账号ID' (Account ID). Below this is a '项目列表' (Project List) table with two rows:

项目ID	项目	所属区域
cn-north-1	cn-north-1	华北-北京一
cn-north-4	cn-north-4	华北-北京四

A red box highlights the '项目ID' column header. A search bar at the top right says '请输入项目名称进行搜索'.

A 修订记录

发布日期	修订记录
2023-09-30	第五次正式发布。 本次更新说明如下： 支持多AZ存储。
2022-12-14	第四次正式发布。 本次更新说明如下： 上线云桌面备份存储库相关参数。
2020-10-29	第三次正式发布。 本次更新说明如下： 上线tag相关接口。
2020-04-01	第二次正式发布。 本次更新说明如下： 删除基于策略的访问控制公测的相关内容。
2019-08-19	第一次正式发布。