

VPC 终端节点

接口参考

文档版本 01
发布日期 2024-12-26



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 使用前必读	1
1.1 概述	1
1.2 调用说明	1
1.3 终端节点 (Endpoint)	1
1.4 约束与限制	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	7
3.3 返回结果	9
4 API	11
4.1 版本管理	11
4.1.1 查询 VPC 终端节点接口版本列表	11
4.1.2 查询指定 VPC 终端节点接口版本信息	15
4.2 终端节点服务功能	20
4.2.1 创建终端节点服务	20
4.2.2 查询终端节点服务列表	33
4.2.3 查询终端节点服务详情	41
4.2.4 修改终端节点服务	47
4.2.5 删除终端节点服务	57
4.2.6 查询连接终端节点服务的连接列表	61
4.2.7 接受或拒绝终端节点连接	65
4.2.8 查询终端节点服务的白名单列表	70
4.2.9 批量添加或移除终端节点服务的白名单	74
4.2.10 查询公共终端节点服务列表	83
4.2.11 查询终端节点服务概要	89
4.2.12 升级终端节点服务	94
4.2.13 修改终端节点服务名称	97
4.2.14 更新终端节点连接描述	101
4.2.15 批量添加终端节点服务的白名单	107
4.2.16 批量删除终端节点服务的白名单	114

4.2.17 更新终端节点服务白名单描述.....	120
4.3 终端节点功能.....	126
4.3.1 创建终端节点.....	126
4.3.2 查询终端节点列表.....	136
4.3.3 查询终端节点详情.....	145
4.3.4 删除终端节点.....	152
4.3.5 更新终端节点.....	156
4.3.6 修改终端节点的路由表.....	164
4.3.7 修改终端节点策略.....	169
4.3.8 删除网关型终端节点策略（待下线）.....	180
4.4 资源配额功能.....	187
4.4.1 查询配额.....	188
4.5 TAG 功能.....	192
4.5.1 查询资源实例接口.....	192
4.5.2 批量添加或删除资源标签接口.....	204
4.5.3 查询租户资源标签接口.....	212
5 应用示例.....	217
5.1 示例：配置跨 VPC 通信的终端节点.....	217
6 附录.....	219
6.1 状态码.....	219
6.2 错误码.....	220
6.3 获取项目 ID.....	233

1 使用前必读

1.1 概述

欢迎使用VPC终端节点（VPC Endpoint）服务。VPC终端节点使您能够将VPC私密地连接到终端节点服务（云服务、用户私有服务），无需弹性公网IP，为您提供性能强大、组网灵活、安全稳定的网络环境。

您可以使用本文档提供的API对VPC终端节点的资源进行相关操作，如创建、查询、修改、删除等。支持的全部操作请参见[2 API概览](#)。

在调用VPC终端节点服务的API之前，请确保已经充分了解VPC终端节点服务的相关概念，详细信息请参见[产品介绍](#)。

1.2 调用说明

VPC终端节点服务提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

同时VPC终端节点服务还提供多种编程语言的SDK供您使用，SDK的使用方法请参见<https://sdkcenter.developer.huaweicloud.com/?product=vpcep>。

1.3 终端节点（Endpoint）

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VPC终端节点的终端节点。

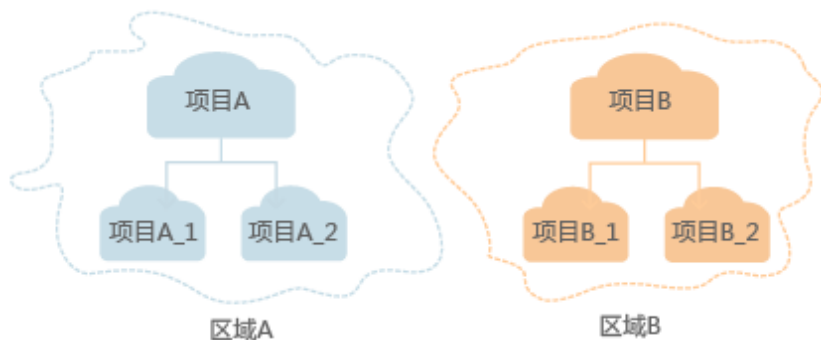
1.4 约束与限制

- 您能创建的VPC终端节点服务资源的数量与配额有关系，如果您想查看服务配额、扩大配额，具体请参见“[配额调整](#)”。
- 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

VPC终端节点提供自研的REST接口。

通过使用VPC终端节点所提供的接口，您可以完整的使用VPC终端节点的所有功能。VPC终端节点主要包括终端节点和终端节点服务两种资源对象。

VPC终端节点提供的具体API如[表2-1](#)所示。

表 2-1 接口说明

子类型	说明
版本管理接口	VPC终端节点API版本查询接口，支持查询所有API或者指定API的版本号。
终端节点服务接口	终端节点服务接口，支持创建、修改、查询、删除终端节点服务、查询列表、查询白名单规则、添加或删除白名单规则、查询连接的终端节点、接受或拒绝终端节点等。 通过这些接口，您可以管理终端节点服务、根据自身业务状况设置规则，来向终端节点提供服务。
终端节点接口	终端节点接口，包括创建、查询、删除终端节点、查询列表，通过这些接口您可以管理终端节点来使用终端节点服务所提供的服务。
资源配额接口	VPC终端节点资源配额查询接口，支持查询VPC终端节点的终端节点服务和终端节点资源的配额。
TAG接口	VPC终端节点的标签管理接口，支持查询资源实例、批量添加或删除资源标签以及查询资源标签。

3 如何调用 API

3.1 构造请求

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

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

请求 URI

请求URI由如下部分组成。

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

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

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

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

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

图 3-1 URI 示意图



说明

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

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务正在请求什么类型的操作。

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

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

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

请求消息头

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

如下公共消息头需要添加到请求中。

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

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

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

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

请求消息体

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

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

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

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

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

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

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

3.2 认证鉴权

调用接口有如下两种认证方式, 您可以选择其中一种进行认证鉴权。

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

AK/SK 认证

📖 说明

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

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

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

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

须知

签名 SDK 只提供签名功能，与服务提供的 SDK 不同，使用时请注意。

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

Token 认证

📖 说明

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

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

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

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

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

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

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

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

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

3.3 返回结果

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码

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

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

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

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

图 3-2 管理员创建 IAM 用户响应消息头

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

响应消息体

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

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

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

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

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

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 版本管理

4.1.1 查询 VPC 终端节点接口版本列表

功能介绍

查询VPC终端节点接口版本列表。

调用方法

请参见[如何调用API](#)。

URI

GET /

请求参数

表 4-1 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-2 响应 Body 参数

参数	参数类型	描述
versions	Array of VersionObject objects	VPC终端节点版本信息列表。

表 4-3 VersionObject

参数	参数类型	描述
status	String	版本状态。 - CURRENT: 表示该版本为主推版本。 - SUPPORT: 表示为老版本, 但是现在还在继续支持。 - DEPRECATED: 表示为废弃版本, 存在后续删除的可能。
id	String	版本ID。 - v1: 当前主推版本
updated	String	版本发布时间。采用UTC时间格式, 格式为: YYYY-MMDDTHH:MM:SSZ
version	String	支持的版本号。
min_version	String	支持的微版本号。若该版本API不支持微版本, 则为空。
links	Array of Link objects	API的URL地址

表 4-4 Link

参数	参数类型	描述
href	String	当前API版本的引用地址。
type	String	发送的实体的MIME类型, 取值为 application/json。
rel	String	当前API版本和被引用地址的关系。

请求示例

查询VPC终端节点接口版本列表

```
GET https://{endpoint}/
```

响应示例

状态码: 200

服务器已成功处理了请求

```
{
  "versions": [ {
    "updated": "2018-09-30T00:00:00Z",
    "version": "1",
    "min_version": "",
    "status": "CURRENT",
    "id": "v1",
    "links": [ {
      "href": "https://{vpcep_uri}/v1",
      "type": "application/json",
      "rel": "self"
    } ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class ListVersionDetailsSolution {

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        ListVersionDetailsRequest request = new ListVersionDetailsRequest();
        try {
            ListVersionDetailsResponse response = client.listVersionDetails(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkvpcep.v1 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
  
    credentials = BasicCredentials(ak, sk)  
  
    client = VpcepClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(VpcepRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListVersionDetailsRequest()  
        response = client.list_version_details(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/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 := vpcep.NewVpcepClient(  
        vpcep.VpcepClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).  
Build()  
  
request := &model.ListVersionDetailsRequest{}  
response, err := client.ListVersionDetails(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.1.2 查询指定 VPC 终端节点接口版本信息

功能介绍

查询指定VPC终端节点接口版本信息。

调用方法

请参见[如何调用API](#)。

URI

GET /{version}

表 4-5 路径参数

参数	是否必选	参数类型	描述
version	是	String	待查询版本号。 取值以v开头，例如v1。若为空，表示查询所有API的版本号。

请求参数

表 4-6 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-7 响应 Body 参数

参数	参数类型	描述
version	VersionObject object	描述VPCEP服务API版本信息列表

表 4-8 VersionObject

参数	参数类型	描述
status	String	版本状态。 - CURRENT：表示该版本为主推版本。 - SUPPORT：表示为老版本，但是现在还在继续支持。 - DEPRECATED：表示为废弃版本，存在后续删除的可能。
id	String	版本ID。 - v1：当前主推版本
updated	String	版本发布时间。采用UTC时间格式，格式为：YYYY-MMDDTHH:MM:SSZ
version	String	支持的版本号。
min_version	String	支持的微版本号。若该版本API不支持微版本，则为空。
links	Array of Link objects	API的URL地址

表 4-9 Link

参数	参数类型	描述
href	String	当前API版本的引用地址。
type	String	发送的实体的MIME类型，取值为application/json。
rel	String	当前API版本和被引用地址的关系。

请求示例

查询指定VPC终端节点接口的v1版本信息。

```
GET https://{endpoint}/v1
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "version":{
    {
      "updated":"2018-09-30T00:00:00Z",
      "version":"1",
      "min_version":"",
      "status":"CURRENT",
      "id":"v1",
      "links":[
        {
          "href":"https://{vpcep_uri}/v1",
          "type":"application/json",
          "rel":"self"
        }
      ]
    }
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class ListSpecifiedVersionDetailsSolution {
```

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

    VpcepClient client = VpcepClient.newBuilder()
        .withCredential(auth)
        .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
        .build();
    ListSpecifiedVersionDetailsRequest request = new ListSpecifiedVersionDetailsRequest();
    request.withVersion("{version}");
    try {
        ListSpecifiedVersionDetailsResponse response = client.listSpecifiedVersionDetails(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.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 = VpcepClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(VpcepRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSpecifiedVersionDetailsRequest()
        request.version = "{version}"
        response = client.list_specified_version_details(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

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

Go

```
package main

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

    request := &model.ListSpecifiedVersionDetailsRequest{}
    request.Version = "{version}"
    response, err := client.ListSpecifiedVersionDetails(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2 终端节点服务功能

4.2.1 创建终端节点服务

功能介绍

创建终端节点服务，允许其他用户创建终端节点连接您创建的终端节点服务，使用您所提供的服务。

该接口为异步接口，调用成功会返回200状态码，说明请求已正常下发。

通常创建终端节点服务需要1~2分钟，可以通过[查询终端节点服务详情](#)查看创建结果。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoint-services

表 4-10 路径参数

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

请求参数

表 4-11 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。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。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-12 请求 Body 参数

参数	是否必选	参数类型	描述
port_id	是	String	标识终端节点服务后端资源的ID， 格式为通用唯一识别码（Universally Unique Identifier，下文简称UUID）。 取值为： <ul style="list-style-type: none">• LB类型：负载均衡器内网IP对应的端口ID。 详细内容请参考《弹性负载均衡API参考》中的“查询负载均衡详情”。• VM类型：弹性云服务器IP地址对应的网卡ID。 详细内容请参考《弹性云服务器API参考》中的“查询云服务器网卡信息”， 详见响应消息中的“port_id”字段。• VIP类型：虚拟IP所在虚拟机的网卡ID（VIP类型业务已不支持，该取值类型已废弃） 说明：<ul style="list-style-type: none">• 创建终端节点服务时，VPC的子网网段不能与198.19.128.0/17重叠。• VPC路由表中自定义路由的目的地址不能与198.19.128.0/17重叠。
service_name	否	String	终端节点服务的名称，长度不大于16，允许传入大小写字母、数字、下划线、中划线。 <ul style="list-style-type: none">• 传入为空，存入值为regionName+.+serviceId• 传入不为空并校验通过，存入值为regionName+.+serviceName+.+serviceId
vpc_id	是	String	终端节点服务对应后端资源所在的VPC的ID。

参数	是否必选	参数类型	描述
approval_enabled	否	Boolean	是否需要审批。 <ul style="list-style-type: none">• false: 不需要审批, 创建的终端节点连接直接为 accepted 状态。• true: 需要审批, 创建的终端节点连接为 pendingAcceptance 状态, 需要终端节点服务所属用户审核后方可使用。 默认为 true, 需要审批。
service_type	否	String	终端节点服务类型。 仅支持将用户私有服务创建为 interface 类型的终端节点服务。 终端节点服务类型包括“网关 (gateway) 型”和“接口 (interface) 型”： <ul style="list-style-type: none">• gateway: 由运维人员配置。用户无需创建, 可直接使用。• interface: 包括运维人员配置的云服务和用户自己创建的私有服务。 其中, 运维人员配置的云服务无需创建, 用户可直接使用。 您可以通过查询公共终端节点服务列表, 查看由运维人员配置的所有用户可见且可连接的终端节点服务, 并通过创建终端节点创建访问 Gateway 和 Interface 类型终端节点服务的终端节点。
server_type	是	String	资源类型。 <ul style="list-style-type: none">• VM: 云服务器, 适用于作为服务器使用。• VIP: 虚拟 IP, 适用于作为虚 IP 场景使用。(该字段已废弃, 请优先使用 LB 类型)• LB: 负载均衡, 适用于高访问量业务和对可靠性和容灾性要求较高的业务。

参数	是否必选	参数类型	描述
ip	否	String	接口型VLAN场景服务端IPv4地址或域名
ports	是	Array of PortList objects	服务开放的端口映射列表，详细内容请参见表4-10。 同一个终端节点服务下，不允许重复的端口映射。若多个终端节点服务共用一个port_id，则终端节点服务之间的所有端口映射的server_port和protocol的组合不能重复， 单次最多添加200个。
tcp_proxy	否	String	用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。 支持携带的客户端信息包括如下两种类型： <ul style="list-style-type: none">• TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。• Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括：<ul style="list-style-type: none">• close：表示关闭代理协议。• toa_open：表示开启代理协议“tcp_toa”。• proxy_open：表示开启代理协议“proxy_protocol”。• open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。默认值为“close”。
tags	否	Array of TagList objects	资源标签列表。同一个终端节点服务最多可添加20个标签。

参数	是否必选	参数类型	描述
description	否	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。 描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
ip_version	否	String	指定终端节点服务的IP版本，仅专业型终端节点服务支持此参数 <ul style="list-style-type: none">• ipv4, IPv4• ipv6, IPv6
snat_network_id	否	String	接口型snat的地址段，ip_version为ipv6时必选。创建服务时使用的VPC内的任意一个网络ID。当服务类型为VIP、VM、ELBV2类型时使用

表 4-13 PortList

参数	是否必选	参数类型	描述
client_port	否	Integer	终端节点访问的端口。 终端节点提供给用户，作为访问终端节点服务的端口，范围1-65535。
server_port	否	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围1-65535。
protocol	否	String	端口映射协议，支持TCP。

表 4-14 TagList

参数	是否必选	参数类型	描述
key	否	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符： _ . : = + - @。 _sys_ 开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	否	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符： _ . : / = + - @。 资源标签值可以为空字符串。

响应参数

状态码： 200

表 4-15 响应 Body 参数

参数	参数类型	描述
id	String	终端节点服务的ID，唯一标识。
port_id	String	标识终端节点服务后端资源的ID，格式为通用唯一识别码（Universally Unique Identifier，下文简称UUID）。 取值为： <ul style="list-style-type: none">LB类型：负载均衡器内网IP对应的端口ID。VM类型：弹性云服务器IP地址对应的网卡ID。VIP类型：虚拟资源所在物理服务器对应的网卡ID。（该字段已废弃，请优先使用LB类型）
service_name	String	终端节点服务的名称。

参数	参数类型	描述
server_type	String	资源类型。 <ul style="list-style-type: none">VM: 云服务器。VIP: 虚拟IP。LB: 增强负载均衡型。
vpc_id	String	终端节点服务对应后端资源所在的VPC的ID。
pool_id	String	终端节点服务对应的集群id
approval_enabled	Boolean	是否需要审批。 <ul style="list-style-type: none">false: 不需要审批, 创建的终端节点连接直接为accepted状态。true: 需要审批, 创建的终端节点连接为pendingAcceptance状态, 需要终端节点服务所属用户审核后方可使用。
status	String	终端节点服务的状态。 <ul style="list-style-type: none">creating: 创建中available: 可连接failed: 失败
service_type	String	终端节点服务类型。 终端节点服务类型包括“网关 (gateway) 型”和“接口 (interface) 型”： <ul style="list-style-type: none">gateway: 由运维人员配置。用户无需创建, 可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。 其中, 运维人员配置的云服务无需创建, 用户可直接使用。 您可以通过创建终端节点创建访问 Gateway和Interface类型终端节点服务的终端节点。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式, 格式为: YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点服务的更新时间。 采用UTC时间格式, 格式为: YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID
ip	String	接口型VLAN场景服务端IPv4地址或域名

参数	参数类型	描述
ports	Array of PortList objects	服务开放的端口映射列表 同一个终端节点服务下，不允许重复的端口映射。 若多个终端节点服务共用一个port_id，则终端节点服务之间的所有端口映射的server_port和protocol的组合不能重复。
tcp_proxy	String	用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。支持携带的客户端信息包括如下两种类型： <ul style="list-style-type: none">• TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。• Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括：<ul style="list-style-type: none">• close：表示关闭代理协议。• toa_open：表示开启代理协议“tcp_toa”。• proxy_open：表示开启代理协议“proxy_protocol”。• open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。默认值为“close”。
tags	Array of TagList objects	资源标签列表
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
enable_policy	Boolean	是否允许自定义终端节点策略。 <ul style="list-style-type: none">• false：不支持设置终端节点策略• true：支持设置终端节点策略 默认为false
ip_version	String	指定终端节点服务的IP版本，仅专业型终端节点服务支持此参数 <ul style="list-style-type: none">• ipv4, IPv4• ipv6, IPv6

表 4-16 PortList

参数	参数类型	描述
client_port	Integer	终端节点访问的端口。 终端节点提供给用户，作为访问终端节点服务的端口，范围1-65535。
server_port	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围1-65535。
protocol	String	端口映射协议，支持TCP。

表 4-17 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

请求示例

创建接口型终端节点服务，资源类型为VM，访问端口为8080和8081，服务端口为90和80，端口映射协议为TCP，无需审批。

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services
{
  "port_id": "4189d3c2-8882-4871-a3c2-d380272eed88",
  "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed80",
  "approval_enabled": false,
  "service_type": "interface",
```



```
"server_type": "VM",  
"ports": [ {  
  "client_port": 8080,  
  "server_port": 90,  
  "protocol": "TCP"  
}, {  
  "client_port": 8081,  
  "server_port": 80,  
  "protocol": "TCP"  
}]  
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{  
  "id": "4189d3c2-8882-4871-a3c2-d380272eed83",  
  "port_id": "4189d3c2-8882-4871-a3c2-d380272eed88",  
  "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed80",  
  "pool_id": "5289d3c2-8882-4871-a3c2-d380272eed80",  
  "status": "available",  
  "approval_enabled": false,  
  "service_name": "test123",  
  "service_type": "interface",  
  "server_type": "VM",  
  "project_id": "6e9dfd51d1124e8d8498dce894923a0d",  
  "created_at": "2022-04-14T09:35:47Z",  
  "ports": [ {  
    "client_port": 8080,  
    "server_port": 90,  
    "protocol": "TCP"  
  }, {  
    "client_port": 8081,  
    "server_port": 80,  
    "protocol": "TCP"  
  } ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建接口型终端节点服务，资源类型为VM，访问端口为8080和8081，服务端口为90和80，端口映射协议为TCP，无需审批。

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.vpcep.v1.region.VpcepRegion;  
import com.huaweicloud.sdk.vpcep.v1.*;  
import com.huaweicloud.sdk.vpcep.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateEndpointServiceSolution {
```

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

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

    VpcepClient client = VpcepClient.newBuilder()
        .withCredential(auth)
        .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateEndpointServiceRequest request = new CreateEndpointServiceRequest();
    CreateEndpointServiceRequestBody body = new CreateEndpointServiceRequestBody();
    List<PortList> listbodyPorts = new ArrayList<>();
    listbodyPorts.add(
        new PortList()
            .withClientPort(8080)
            .withServerPort(90)
            .withProtocol(PortList.ProtocolEnum.fromValue("TCP"))
    );
    listbodyPorts.add(
        new PortList()
            .withClientPort(8081)
            .withServerPort(80)
            .withProtocol(PortList.ProtocolEnum.fromValue("TCP"))
    );
    body.withPorts(listbodyPorts);
    body.withServerType(CreateEndpointServiceRequestBody.ServerTypeEnum.fromValue("VM"));
    body.withServiceType(CreateEndpointServiceRequestBody.ServiceTypeEnum.fromValue("interface"));
    body.withApprovalEnabled(false);
    body.withVpclid("4189d3c2-8882-4871-a3c2-d380272eed80");
    body.withPortId("4189d3c2-8882-4871-a3c2-d380272eed88");
    request.withBody(body);
    try {
        CreateEndpointServiceResponse response = client.createEndpointService(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

创建接口型终端节点服务，资源类型为VM，访问端口为8080和8081，服务端口为90和80，端口映射协议为TCP，无需审批。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpc.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateEndpointServiceRequest()
        listPortsbody = [
            PortList(
                client_port=8080,
                server_port=90,
                protocol="TCP"
            ),
            PortList(
                client_port=8081,
                server_port=80,
                protocol="TCP"
            )
        ]
        request.body = CreateEndpointServiceRequestBody(
            ports=listPortsbody,
            server_type="VM",
            service_type="interface",
            approval_enabled=False,
            vpc_id="4189d3c2-8882-4871-a3c2-d380272eed80",
            port_id="4189d3c2-8882-4871-a3c2-d380272eed88"
        )
        response = client.create_endpoint_service(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建接口型终端节点服务，资源类型为VM，访问端口为8080和8081，服务端口为90和80，端口映射协议为TCP，无需审批。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

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

client := vpcep.NewVpcepClient(
    vpcep.VpcepClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateEndpointServiceRequest{
    clientPortPorts:= int32(8080)
    serverPortPorts:= int32(90)
    protocolPorts:= model.GetPortListProtocolEnum().TCP
    clientPortPorts1:= int32(8081)
    serverPortPorts1:= int32(80)
    protocolPorts1:= model.GetPortListProtocolEnum().TCP
    var listPortsbody = []model.PortList{
        {
            ClientPort: &clientPortPorts,
            ServerPort: &serverPortPorts,
            Protocol: &protocolPorts,
        },
        {
            ClientPort: &clientPortPorts1,
            ServerPort: &serverPortPorts1,
            Protocol: &protocolPorts1,
        },
    }
    serviceTypeCreateEndpointServiceRequestBody:=
model.GetCreateEndpointServiceRequestBodyServiceTypeEnum().INTERFACE
approvalEnabledCreateEndpointServiceRequestBody:= false
request.Body = &model.CreateEndpointServiceRequestBody{
    Ports: listPortsbody,
    ServerType: model.GetCreateEndpointServiceRequestBodyServerTypeEnum().VM,
    ServiceType: &serviceTypeCreateEndpointServiceRequestBody,
    ApprovalEnabled: &approvalEnabledCreateEndpointServiceRequestBody,
    Vpclid: "4189d3c2-8882-4871-a3c2-d380272eed80",
    PortId: "4189d3c2-8882-4871-a3c2-d380272eed88",
}
response, err := client.CreateEndpointService(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.2 查询终端节点服务列表

功能介绍

查询当前用户下的终端节点服务的列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services

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
endpoint_service_name	否	String	终端节点服务的名称，支持大小写，前后模糊匹配。
id	否	String	终端节点服务的ID，唯一标识。
status	否	String	终端节点服务的状态。 <ul style="list-style-type: none">creating：创建中available：可连接failed：失败deleting：删除中
sort_key	否	String	查询结果中终端节点服务列表的排序字段，取值为： <ul style="list-style-type: none">create_at：终端节点服务的创建时间update_at：终端节点服务的更新时间 默认值为create_at。

参数	是否必选	参数类型	描述
sort_dir	否	String	查询结果中终端节点服务列表的排序方式，取值为： <ul style="list-style-type: none">• desc：降序排序• asc：升序排序 默认值为desc。
limit	否	Integer	查询返回的终端节点服务数量限制，即每页返回的终端节点服务的个数。 取值范围：0~1000，取值一般为10，20或者50，默认为10。
offset	否	Integer	偏移量。 偏移量为一个大于0小于终端节点服务总个数的整数， 表示从偏移量后面的终端节点服务开始查询。
public_border_group	否	String	筛选结果中匹配边缘属性的EPS
net_type	否	String	后端类型

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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-21 响应 Body 参数

参数	参数类型	描述
endpoint_services	Array of ServiceList objects	终端节点服务列表
total_count	Integer	满足查询条件的终端节点服务总条数，不受分页（即limit、offset参数）影响。

表 4-22 ServiceList

参数	参数类型	描述
id	String	终端节点服务的ID，唯一标识。
port_id	String	标识终端节点服务后端资源的ID，格式为通用唯一识别码（Universally Unique Identifier，下文简称UUID）。取值为： <ul style="list-style-type: none">LB类型：负载均衡器内网IP对应的端口ID。VM类型：弹性云服务器IP地址对应的网卡ID。VIP类型：虚拟资源所在物理服务器对应的网卡ID。（该字段已废弃，请优先使用LB类型）
service_name	String	终端节点服务的名称。
server_type	String	资源类型。 <ul style="list-style-type: none">VM：云服务器。VIP：虚拟IP。LB：增强负载均衡型。
vpc_id	String	终端节点服务对应后端资源所在的VPC的ID。
approval_enabled	Boolean	是否需要审批。 <ul style="list-style-type: none">false：不需要审批，创建的终端节点连接直接为accepted状态。true：需要审批，创建的终端节点连接为pendingAcceptance状态，需要终端节点服务所属用户审核后后方可使用。

参数	参数类型	描述
status	String	终端节点服务的状态。 <ul style="list-style-type: none">creating: 创建中available: 可连接failed: 失败deleting: 删除中
service_type	String	终端节点服务类型。 终端节点服务类型包括“网关（gateway）型”和“接口（interface）型”： <ul style="list-style-type: none">gateway: 由运维人员配置。用户无需创建，可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。 其中，运维人员配置的云服务无需创建，用户可直接使用。 您可以通过创建终端节点创建访问Gateway和Interface类型终端节点服务的终端节点。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点服务的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
domain_id	String	Domain ID
ports	Array of PortList objects	服务开放的端口映射列表 同一个终端节点服务下，不允许重复的端口映射。 若多个终端节点服务共用一个port_id，则终端节点服务之间的所有端口映射的server_port和protocol的组合不能重复。
ip	String	接口型VLAN场景服务端IPv4地址或域名
tags	Array of TagList objects	资源标签列表
connection_count	Integer	终端节点服务下连接的状态为“创建中”或“已接受”的终端节点的个数。

参数	参数类型	描述
tcp_proxy	String	<p>用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。支持携带的客户端信息包括如下两种类型：</p> <ul style="list-style-type: none"> TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。 Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括： <ul style="list-style-type: none"> close：表示关闭代理协议。 toa_open：表示开启代理协议“tcp_toa”。 proxy_open：表示开启代理协议“proxy_protocol”。 open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。 <p>默认值为“close”。</p>
error	Array of Error objects	提交任务异常时返回的异常信息
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
public_border_group	String	终端节点服务对应Pool的Public Border Group信息
enable_policy	Boolean	<p>是否允许自定义终端节点策略。</p> <ul style="list-style-type: none"> false：不支持设置终端节点策略 true：支持设置终端节点策略 <p>默认为false</p>

表 4-23 PortList

参数	参数类型	描述
client_port	Integer	<p>终端节点访问的端口。</p> <p>终端节点提供给用户，作为访问终端节点服务的端口，范围1-65535。</p>

参数	参数类型	描述
server_port	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围1-65535。
protocol	String	端口映射协议，支持TCP。

表 4-24 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_ 开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-25 Error

参数	参数类型	描述
message	String	任务异常错误信息描述
code	String	任务异常错误信息编码

请求示例

查询终端节点服务列表

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services?  
endpoint_service_name={endpoint_service_name}&id={id}&sort_key={sort_key}&sort_dir={sort_dir}&limit={li  
mit}&offset={offset}&status={status}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "endpoint_services": [ {
    "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
    "port_id": "4189d3c2-8882-4871-a3c2-d380272eed88",
    "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed80",
    "status": "available",
    "approval_enabled": false,
    "service_name": "test123",
    "server_type": "VM",
    "service_type": "interface",
    "ports": [ {
      "client_port": "8080",
      "server_port": "80",
      "protocol": "TCP"
    }, {
      "client_port": "8081",
      "server_port": "80",
      "protocol": "TCP"
    } ],
    "project_id": "6e9dfd51d1124e8d8498dce894923a0d",
    "created_at": "2022-04-14T09:35:47Z",
    "updated_at": "2022-04-14T09:36:47Z",
    "public_border_group": "br-abc-aaa1"
  } ],
  "total_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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class ListEndpointServiceSolution {

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

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

```
VpcepClient client = VpcepClient.newBuilder()
    .withCredential(auth)
    .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
    .build();
ListEndpointServiceRequest request = new ListEndpointServiceRequest();
try {
    ListEndpointServiceResponse response = client.listEndpointService(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.3 查询终端节点服务详情

功能介绍

查询终端节点服务的详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}

表 4-26 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-27 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-28 响应 Body 参数

参数	参数类型	描述
id	String	终端节点服务的ID，唯一标识。

参数	参数类型	描述
port_id	String	标识终端节点服务后端资源的ID，格式为通用唯一识别码（Universally Unique Identifier，下文简称UUID）。 取值为： <ul style="list-style-type: none">LB类型：负载均衡器内网IP对应的端口ID。VM类型：弹性云服务器IP地址对应的网卡ID。VIP类型：虚拟资源所在物理服务器对应的网卡ID。（该字段已废弃，请优先使用LB类型）
service_name	String	终端节点服务的名称。
server_type	String	资源类型。 <ul style="list-style-type: none">VM：云服务器。VIP：虚拟IP。LB：增强负载均衡型。
vpc_id	String	终端节点服务对应后端资源所在的VPC的ID。
approval_enabled	Boolean	是否需要审批。 <ul style="list-style-type: none">false：不需要审批，创建的终端节点连接直接为accepted状态。true：需要审批，创建的终端节点连接为pendingAcceptance状态，需要终端节点服务所属用户审核后方可使用。
status	String	终端节点服务的状态。 <ul style="list-style-type: none">creating：创建中available：可连接failed：失败deleting：删除中

参数	参数类型	描述
service_type	String	终端节点服务类型。 终端节点服务类型包括“网关（gateway）型”和“接口（interface）型”： <ul style="list-style-type: none">gateway：由运维人员配置。用户无需创建，可直接使用。interface：包括运维人员配置的云服务和用户自己创建的私有服务。 其中，运维人员配置的云服务无需创建，用户可直接使用。 您可以通过创建终端节点创建访问Gateway和Interface类型终端节点服务的终端节。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点服务的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
cidr_type	String	网段类型。 public：公网网段 internal：内网网段 默认值为internal。
ip	String	接口型VLAN场景服务端IPv4地址或域名
ports	Array of PortList objects	服务开放的端口映射列表，详细内容请参见表4-17 同一个终端节点服务下，不允许重复的端口映射。若多个终端节点服务共用一个port_id，则 终端节点服务之间的所有端口映射的server_port和protocol的组合不能重复。

参数	参数类型	描述
tcp_proxy	String	<p>用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。支持携带的客户端信息包括如下两种类型：</p> <ul style="list-style-type: none"> • TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。 • Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括： <ul style="list-style-type: none"> • close：表示关闭代理协议。 • toa_open：表示开启代理协议“tcp_toa”。 • proxy_open：表示开启代理协议“proxy_protocol”。 • open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。 <p>默认值为“close”。</p>
tags	Array of TagList objects	资源标签列表
error	Array of Error objects	提交任务异常时返回的异常信息
enable_policy	Boolean	<p>是否允许自定义终端节点策略。</p> <ul style="list-style-type: none"> • false：不支持设置终端节点策略 • true：支持设置终端节点策略 <p>默认为false</p>
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。

表 4-29 PortList

参数	参数类型	描述
client_port	Integer	<p>终端节点访问的端口。</p> <p>终端节点提供给用户，作为访问终端节点服务的端口，范围1-65535。</p>

参数	参数类型	描述
server_port	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围1-65535。
protocol	String	端口映射协议，支持TCP。

表 4-30 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_ 开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-31 Error

参数	参数类型	描述
message	String	任务异常错误信息描述
code	String	任务异常错误信息编码

请求示例

查询终端节点服务详情

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}
```

查询ID为“4189d3c2-8882-4871-a3c2-d380272eed88”的终端节点服务详情。GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
  "port_id": "4189d3c2-8882-4871-a3c2-d380272eed88",
  "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed80",
  "status": "available",
  "approval_enabled": false,
  "service_name": "test123",
  "server_type": "VM",
  "service_type": "interface",
  "ports": [ {
    "client_port": "8080",
    "server_port": "80",
    "protocol": "TCP"
  }, {
    "client_port": "8081",
    "server_port": "80",
    "protocol": "TCP"
  } ],
  "project_id": "6e9dfd51d1124e8d8498dce894923a0d",
  "created_at": "2022-04-14T09:35:47Z",
  "updated_at": "2022-04-14T09:36:47Z"
}
```

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.4 修改终端节点服务

功能介绍

修改终端节点服务。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}

表 4-32 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID

请求参数

表 4-33 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-34 请求 Body 参数

参数	是否必选	参数类型	描述
approval_enabled	否	Boolean	是否需要审批。 <ul style="list-style-type: none">• false: 不需审批，创建的终端节点连接直接为accepted状态。• true: 需审批，创建的终端节点连接需要终端节点服务所属用户审核后方可使用。 默认为true，需要审批。
service_name	否	String	终端节点服务的名称，长度不大于16，允许传入大小写字母、数字、下划线、中划线。

参数	是否必选	参数类型	描述
ports	否	Array of PortList objects	服务开放的端口映射列表，同一个终端节点服务下，不允许重复的端口映射。若多个终端节点服务共用一个port_id，则终端节点之间服务的所有端口映射的server_port和protocol的组合不能重复，单次最多添加200个。该参数值将被全量更新。
port_id	否	String	弹性云服务器IP地址对应的网卡ID
tcp_proxy	否	String	用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。 支持携带的客户端信息包括如下两种类型： <ul style="list-style-type: none">• TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。• Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括：<ul style="list-style-type: none">• close：表示关闭代理协议。• toa_open：表示开启代理协议“tcp_toa”。• proxy_open：表示开启代理协议“proxy_protocol”。• open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。默认值为“close”。
description	否	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
ip	否	String	接口型VLAN场景服务端IPv4地址或域名

表 4-35 PortList

参数	是否必选	参数类型	描述
client_port	否	Integer	终端节点访问的端口。 终端节点提供给用户，作为访问终端节点服务的端口，范围 1-65535。
server_port	否	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围 1-65535。
protocol	否	String	端口映射协议，支持TCP。

响应参数

状态码： 200

表 4-36 响应 Body 参数

参数	参数类型	描述
id	String	终端节点服务的ID，唯一标识。
port_id	String	标识终端节点服务后端资源的ID，格式为通用唯一识别码（Universally Unique Identifier，下文简称UUID）。 取值为： <ul style="list-style-type: none">LB类型：负载均衡器内网IP对应的端口ID。VM类型：弹性云服务器IP地址对应的网卡ID。VIP类型：虚拟资源所在物理服务器对应的网卡ID。（该字段已废弃，请优先使用LB类型）
service_name	String	终端节点服务的名称。
server_type	String	资源类型。 <ul style="list-style-type: none">VM：云服务器。VIP：虚拟IP。LB：增强负载均衡型。
vpc_id	String	终端节点服务对应后端资源所在的VPC的ID。
pool_id	String	终端节点服务对应的集群id

参数	参数类型	描述
approval_enabled	Boolean	是否需要审批。 <ul style="list-style-type: none">• false: 不需要审批, 创建的终端节点连接直接为accepted状态。• true: 需要审批, 创建的终端节点连接为pendingAcceptance状态, 需要终端节点服务所属用户审核后方可使用。
status	String	终端节点服务的状态。 <ul style="list-style-type: none">• creating: 创建中• available: 可连接• failed: 失败
service_type	String	终端节点服务类型。 终端节点服务类型包括“网关 (gateway) 型”和“接口 (interface) 型”： <ul style="list-style-type: none">• gateway: 由运维人员配置。用户无需创建, 可直接使用。• interface: 包括运维人员配置的云服务和用户自己创建的私有服务。 其中, 运维人员配置的云服务无需创建, 用户可直接使用。 您可以通过创建终端节点创建访问 Gateway和Interface类型终端节点服务的终端节点。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式, 格式为: YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点服务的更新时间。 采用UTC时间格式, 格式为: YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID
ip	String	接口型VLAN场景服务端IPv4地址或域名
ports	Array of PortList objects	服务开放的端口映射列表 同一个终端节点服务下, 不允许重复的端口映射。 若多个终端节点服务共用一个port_id, 则终端节点服务之间的所有端口映射的server_port和protocol的组合不能重复。

参数	参数类型	描述
tcp_proxy	String	<p>用于控制将哪些信息（如客户端的源IP、源端口、marker_id等）携带到服务端。支持携带的客户端信息包括如下两种类型：</p> <ul style="list-style-type: none">• TCP TOA：表示将客户端信息插入到tcp option字段中携带至服务端。 说明：仅当后端资源为OBS时，支持TCP TOA类型信息携带方式。• Proxy Protocol：表示将客户端信息插入到tcp payload字段中携带至服务端。 仅当服务端支持解析上述字段时，该参数设置才有效。 该参数的取值包括：<ul style="list-style-type: none">• close：表示关闭代理协议。• toa_open：表示开启代理协议“tcp_toa”。• proxy_open：表示开启代理协议“proxy_protocol”。• open：表示同时开启代理协议“tcp_toa”和“proxy_protocol”。默认值为“close”。
tags	Array of TagList objects	资源标签列表
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
enable_policy	Boolean	<p>是否允许自定义终端节点策略。</p> <ul style="list-style-type: none">• false：不支持设置终端节点策略• true：支持设置终端节点策略 默认为false
ip_version	String	<p>指定终端节点服务的IP版本，仅专业型终端节点服务支持此参数</p> <ul style="list-style-type: none">• ipv4, IPv4• ipv6, IPv6

表 4-37 PortList

参数	参数类型	描述
client_port	Integer	终端节点访问的端口。 终端节点提供给用户，作为访问终端节点服务的端口，范围1-65535。
server_port	Integer	终端节点服务的端口。 终端节点服务绑定了后端资源，作为提供服务的端口，范围1-65535。
protocol	String	端口映射协议，支持TCP。

表 4-38 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

请求示例

修改终端节点服务，访问端口为8081和8082，服务端口为22和23，需审批。

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88
```

```
{
  "approval_enabled": true,
  "service_name": "test",
  "ports": [{
    "client_port": 8081,
    "server_port": 22,
    "protocol": "TCP"
  }, {
    "client_port": 8082,
    "server_port": 23,
```

```
"protocol" : "TCP"  
}]  
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{  
  "id" : "4189d3c2-8882-4871-a3c2-d380272eed83",  
  "port_id" : "4189d3c2-8882-4871-a3c2-d380272eed88",  
  "vpc_id" : "4189d3c2-8882-4871-a3c2-d380272eed80",  
  "pool_id" : "5289d3c2-8882-4871-a3c2-d380272eed80",  
  "status" : "available",  
  "approval_enabled" : false,  
  "service_name" : "test123",  
  "service_type" : "interface",  
  "server_type" : "VM",  
  "project_id" : "6e9dfd51d1124e8d8498dce894923a0d",  
  "created_at" : "2022-04-14T09:35:47Z",  
  "ports" : [{  
    "client_port" : 8080,  
    "server_port" : 90,  
    "protocol" : "TCP"  
  }, {  
    "client_port" : 8081,  
    "server_port" : 80,  
    "protocol" : "TCP"  
  }],  
  "tcp_proxy" : "proxy_open"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改终端节点服务，访问端口为8081和8082，服务端口为22和23，需审批。

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

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

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

VpcepClient client = VpcepClient.newBuilder()
    .withCredential(auth)
    .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateEndpointServiceRequest request = new UpdateEndpointServiceRequest();
request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
UpdateEndpointServiceRequestBody body = new UpdateEndpointServiceRequestBody();
List<PortList> listbodyPorts = new ArrayList<>();
listbodyPorts.add(
    new PortList()
        .withClientPort(8081)
        .withServerPort(22)
        .withProtocol(PortList.ProtocolEnum.fromValue("TCP"))
);
listbodyPorts.add(
    new PortList()
        .withClientPort(8082)
        .withServerPort(23)
        .withProtocol(PortList.ProtocolEnum.fromValue("TCP"))
);
body.withPorts(listbodyPorts);
body.withServiceName("test");
body.withApprovalEnabled(true);
request.withBody(body);
try {
    UpdateEndpointServiceResponse response = client.updateEndpointService(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

修改终端节点服务，访问端口为8081和8082，服务端口为22和23，需审批。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

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

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

try:
    request = UpdateEndpointServiceRequest()
    request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
    listPortsbody = [
        PortList(
            client_port=8081,
            server_port=22,
            protocol="TCP"
        ),
        PortList(
            client_port=8082,
            server_port=23,
            protocol="TCP"
        )
    ]
    request.body = UpdateEndpointServiceRequestBody(
        ports=listPortsbody,
        service_name="test",
        approval_enabled=True
    )
    response = client.update_endpoint_service(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

修改终端节点服务，访问端口为8081和8082，服务端口为22和23，需审批。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.UpdateEndpointServiceRequest{
request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
clientPortPorts:= int32(8081)
serverPortPorts:= int32(22)
protocolPorts:= model.GetPortListProtocolEnum().TCP
clientPortPorts1:= int32(8082)
serverPortPorts1:= int32(23)
protocolPorts1:= model.GetPortListProtocolEnum().TCP
var listPortsbody = []model.PortList{
    {
        ClientPort: &clientPortPorts,
        ServerPort: &serverPortPorts,
        Protocol: &protocolPorts,
    },
    {
        ClientPort: &clientPortPorts1,
        ServerPort: &serverPortPorts1,
        Protocol: &protocolPorts1,
    },
}
serviceNameUpdateEndpointServiceRequestBody:= "test"
approvalEnabledUpdateEndpointServiceRequestBody:= true
request.Body = &model.UpdateEndpointServiceRequestBody{
    Ports: &listPortsbody,
    ServiceName: &serviceNameUpdateEndpointServiceRequestBody,
    ApprovalEnabled: &approvalEnabledUpdateEndpointServiceRequestBody,
}
response, err := client.UpdateEndpointService(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.5 删除终端节点服务

功能介绍

删除终端节点服务。

该接口为异步接口，调用成功会返回200状态码，说明请求已正常下发。通常删除终端节点服务需要1~2分钟，可以通过[查询终端节点服务详情](#)查看删除结果。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}

表 4-39 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-40 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

无

请求示例

删除终端节点服务

```
DELETE https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88
```

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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class DeleteEndpointServiceSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteEndpointServiceRequest request = new DeleteEndpointServiceRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        try {
            DeleteEndpointServiceResponse response = client.deleteEndpointService(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

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

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = DeleteEndpointServiceRequest()
    request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
    response = client.delete_endpoint_service(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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


更多

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

状态码

状态码	描述
204	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.6 查询连接终端节点服务的连接列表

功能介绍

查询连接当前用户下的某一个终端节点服务的连接列表。marker_id是连接的唯一标识。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/connections

表 4-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

表 4-42 Query 参数

参数	是否必选	参数类型	描述
id	否	String	终端节点的ID，唯一标识。
marker_id	否	String	终端节点的报文标识。

参数	是否必选	参数类型	描述
status	否	String	终端节点的连接状态。 <ul style="list-style-type: none">• pendingAcceptance: 待接受• accepted: 已接受• rejected: 已拒绝• failed: 失败
sort_key	否	String	查询结果中终端节点列表的排序字段，取值为： <ul style="list-style-type: none">• create_at: 终端节点的创建时间• update_at: 终端节点的更新时间 默认值为create_at。
sort_dir	否	String	查询结果中终端节点列表的排序方式，取值为： <ul style="list-style-type: none">• desc: 降序排序• asc: 升序排序 默认值为desc。
limit	否	Integer	查询返回终端节点服务的连接列表限制每页个数，即每页返回的个数。 取值范围：0~1000，取值一般为10，20或者50，默认为10。
offset	否	Integer	偏移量。 偏移量为一个大于0小于终端节点服务总个数的整数， 表示从偏移量后面的终端节点服务开始查询。

请求参数

表 4-43 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)。

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-44 响应 Body 参数

参数	参数类型	描述
connections	Array of ConnectionEndpoints objects	连接列表。
total_count	Integer	满足查询条件的终端节点总条数，不受分页（即limit、offset参数）影响。

表 4-45 ConnectionEndpoints

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
marker_id	Integer	终端节点的报文标识。
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
domain_id	String	用户的Domain ID。
error	Array of QueryError objects	错误信息。 当终端节点服务状态异常，即“status”的值为“failed”时，会返回该字段。

参数	参数类型	描述
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
description	String	终端节点连接描述。

表 4-46 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

请求示例

查询连接终端节点服务的连接列表

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/connections?
id={vpc_endpoint_id}&marker_id={marker_id}&status={status}&sort_key={sort_key }&sort_dir={sort_dir}&limi
t={limit}&offset={offset}
```

查询连接ID为“4189d3c2-8882-4871-a3c2-d380272eed88”的终端节点服务连接列表。GET https://
{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/connections

响应示例

状态码: 200

服务器已成功处理了请求

```
{
  "connections": [ {
    "id": "adb7b229-bb11-4072-bcc0-3327cd784263",
    "status": "accepted",
    "marker_id": "16777510",
    "domain_id": "5fc973eea581490997e82ea11a1df31f",
    "created_at": "2018-09-17T11:10:11Z",
    "updated_at": "2018-09-17T11:10:12Z"
  }, {
    "id": "fd69d29f-dc29-4a9b-80d8-b51d1e7e58ea",
    "status": "accepted",
    "marker_id": "16777513",
    "domain_id": "5fc973eea581490997e82ea11a1df31f",
    "created_at": "2018-09-17T07:28:56Z",
    "updated_at": "2018-09-17T07:28:58Z"
  } ],
  "total_count": 2
}
```

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.7 接受或拒绝终端节点的连接

功能介绍

接受或者拒绝终端节点连接到当前的终端节点服务。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/connections/action

表 4-47 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-48 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)。

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-49 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	允许或拒绝连接。 <ul style="list-style-type: none">• receive: 允许连接。• reject: 拒绝连接。
endpoints	是	Array of strings	终端节点ID列表。 每次请求目前支持单条endpoint的接受或拒绝。

响应参数

状态码： 200

表 4-50 响应 Body 参数

参数	参数类型	描述
connections	Array of ConnectionEndpoints objects	连接列表

表 4-51 ConnectionEndpoints

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
marker_id	Integer	终端节点的报文标识。
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ

参数	参数类型	描述
domain_id	String	用户的Domain ID。
error	Array of QueryError objects	错误信息。 当终端节点服务状态异常，即“status”的值为“failed”时，会返回该字段。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
description	String	终端节点连接描述。

表 4-52 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

请求示例

接受终端节点连接

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/connections/action
```

```
{
  "endpoints": [ "705290f3-0d00-41f2-aedc-71f09844e879" ],
  "action": "receive"
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "connections": [ {
    "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
    "status": "accepted",
    "marker_id": 422321321312321321,
    "domain_id": "6e9dfd51d1124e8d8498dce894923a0d",
    "created_at": "2022-04-14T09:35:47Z",
    "updated_at": "2022-04-14T09:36:47Z"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

接受终端节点的连接

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

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

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

public class AcceptOrRejectEndpointSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        AcceptOrRejectEndpointRequest request = new AcceptOrRejectEndpointRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        AcceptOrRejectEndpointRequestBody body = new AcceptOrRejectEndpointRequestBody();
        List<String> listbodyEndpoints = new ArrayList<>();
        listbodyEndpoints.add("705290f3-0d00-41f2-aedc-71f09844e879");
        body.withEndpoints(listbodyEndpoints);
        body.withAction(AcceptOrRejectEndpointRequestBody.ActionEnum.fromValue("receive"));
        request.withBody(body);
        try {
            AcceptOrRejectEndpointResponse response = client.acceptOrRejectEndpoint(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

接受终端节点的连接

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AcceptOrRejectEndpointRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listEndpointsbody = [
            "705290f3-0d00-41f2-aedc-71f09844e879"
        ]
        request.body = AcceptOrRejectEndpointRequestBody(
            endpoints=listEndpointsbody,
            action="receive"
        )
        response = client.accept_or_reject_endpoint(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

接受终端节点的连接

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

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

client := vpcep.NewVpcepClient(
    vpcep.VpcepClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AcceptOrRejectEndpointRequest{}
request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
var listEndpointsbody = []string{
    "705290f3-0d00-41f2-aedc-71f09844e879",
}
request.Body = &model.AcceptOrRejectEndpointRequestBody{
    Endpoints: listEndpointsbody,
    Action: model.GetAcceptOrRejectEndpointRequestBodyActionEnum().RECEIVE,
}
response, err := client.AcceptOrRejectEndpoint(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.8 查询终端节点服务的白名单列表

功能介绍

查询当前用户下终端节点服务的白名单列表。

说明

本账号默认在当前用户下终端节点服务的白名单中。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions

表 4-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

表 4-54 Query 参数

参数	是否必选	参数类型	描述
permission	否	String	权限账号ID，格式为“iam:domain::domain_id”。其中“domain_id”为授权用户的账号ID，例如“iam:domain::6e9dfd51d1124e8d8498dce894923a0d”，支持模糊搜索。
limit	否	Integer	查询返回终端节点服务的白名单数量限制，即每页返回的个数。取值范围：0~500，取值一般为10，20或者50，默认为10。
offset	否	Integer	偏移量。偏移量为一个大于0小于终端节点服务总个数的整数，表示从偏移量后面的终端节点服务开始查询。
sort_key	否	String	查询结果中终端节点服务列表的排序字段，取值为： <ul style="list-style-type: none">• create_at：终端节点服务的创建时间• update_at：终端节点服务的更新时间 默认值为create_at。

参数	是否必选	参数类型	描述
sort_dir	否	String	查询结果中白名单列表的排序方式，取值为： <ul style="list-style-type: none">• desc：降序排序• asc：升序排序 默认值为desc。

请求参数

表 4-55 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-56 响应 Body 参数

参数	参数类型	描述
permissions	Array of PermissionObject objects	permission列表。
total_count	Integer	满足查询条件的终端节点服务的白名单总条数，不受分页（即limit、offset参数）影响。

表 4-57 PermissionObject

参数	参数类型	描述
id	String	permission的ID，唯一标识。

参数	参数类型	描述
permission	String	<p>permission列表。</p> <p>权限格式为：</p> <ul style="list-style-type: none"> iam:domain::domain_id。其中： <ul style="list-style-type: none"> “iam:domain::”为固定格式， “domain_id”为可连接用户的账号ID。 domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“*”，最大长度可以传64。 <ul style="list-style-type: none"> organizations:orgPath::org_path。其中： <ul style="list-style-type: none"> “organizations:orgPath::”为固定格式，org_path为可连接用户的组织路径。 org_path类型支持“a~z”、“A~Z”、“0~9”、“/ - ?”或者“*”，最大长度可以传1024。 “*”表示所有终端节点可连接。 <p>示例：</p> <ul style="list-style-type: none"> iam:domain::6e9dfd51d1124e8d8498dce894923a0dd organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rlbu1vmxdw5ahdkknxvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*
permission_type	String	<p>终端节点服务白名单类型。</p> <ul style="list-style-type: none"> domainId：基于账户ID配置终端节点服务白名单。 orgPath：基于账户所在组织路径配置终端节点服务白名单。
created_at	String	<p>白名单的添加时间。</p> <p>采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ</p>

请求示例

查询终端节点服务的白名单列表

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions?permission={permission}&sort_dir={sort_dir}&limit={limit}&offset={offset}
```

查询终端节点服务ID为“4189d3c2-8882-4871-a3c2-d380272eed88”的白名单列表。GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "permissions": [ {
    "id": "f2659906-c622-480a-83e9-ef42bdb67b90",
    "permission": "*",
    "created_at": "2018-10-18T13:26:40Z"
  } ],
  "total_count": 1
}
```

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.9 批量添加或删除终端节点服务的白名单

功能介绍

批量添加或删除当前用户下终端节点服务的白名单。

说明

本账号默认在自身用户的终端节点服务的白名单中。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions/action

表 4-58 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-59 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-60 请求 Body 参数

参数	是否必选	参数类型	描述
permissions	是	Array of strings	<p>permission列表。 权限格式为：</p> <ul style="list-style-type: none"> iam:domain::domain_id。其中： “iam:domain::”为固定格式，“domain_id”为可连接用户的账号ID。 domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“*”，最大长度可以传64。 organizations:orgPath::org_path。其中： “organizations:orgPath::”为固定格式，org_path为可连接用户的组织路径。 org_path类型支持“a~z”、“A~Z”、“0~9”、“/-?”或者“*”，最大长度可以传1024。 “*”表示所有终端节点可连接。 <p>示例：</p> <ul style="list-style-type: none"> iam:domain::6e9dfd51d1124e8d8498dce894923a0dd organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltaws m6qct5z7rkvl/*
permission_type	否	String	<p>终端节点服务白名单类型。</p> <ul style="list-style-type: none"> domainId：基于账户ID配置终端节点服务白名单。 orgPath：基于账户所在组织路径配置终端节点服务白名单。
action	是	String	<p>要执行的操作。 add/remove。</p>

响应参数

状态码： 200

表 4-61 响应 Body 参数

参数	参数类型	描述
permissions	Array of strings	<p>permission列表。</p> <p>权限格式为：</p> <ul style="list-style-type: none"> iam:domain::domain_id。其中： “iam:domain::”为固定格式， “domain_id”为可连接用户的账号ID。 domain_id类型支持输入包括 “a~z”、“A~Z”、“0~9”或者 “*”，最大长度可以传64。 organizations:orgPath::org_path。其中： “organizations:orgPath::”为固定格式，org_path为可连接用户的组织路径。 org_path类型支持“a~z”、“A~Z”、“0~9”、“/-?”或者 “*”，最大长度可以传1024。 “*”表示所有终端节点可连接。 <p>示例：</p> <ul style="list-style-type: none"> iam:domain::6e9dfd51d1124e8d8498dce894923a0dd organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldb1vmxdw5ahdkknxnv5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*
permission_type	String	<p>终端节点服务白名单类型。</p> <ul style="list-style-type: none"> domainId：基于账户ID配置终端节点服务白名单。 orgPath：基于账户所在组织路径配置终端节点服务白名单。

请求示例

- 添加终端节点服务的白名单

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions/action
```

```
{
```

```
"permissions" : [ "iam:domain::fc973eea581490997e82ea11a1d0101" ],  
"action" : "add"  
}
```

- 移除终端节点服务的白名单

POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions/action

```
{  
  "permissions" : [ "iam:domain::fc973eea581490997e82ea11a1d0101" ],  
  "action" : "remove"  
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{  
  "permissions" : [ "iam:domain::5fc973eea581490997e82ea11a1d0101",  
    "iam:domain::5fc973eea581490997e82ea11a1d0102" ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 添加终端节点服务的白名单

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.vpcep.v1.region.VpcepRegion;  
import com.huaweicloud.sdk.vpcep.v1.*;  
import com.huaweicloud.sdk.vpcep.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class AddOrRemoveServicePermissionsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        VpcepClient client = VpcepClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        AddOrRemoveServicePermissionsRequest request = new
AddOrRemoveServicePermissionsRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        AddOrRemoveServicePermissionsRequestBody body = new
AddOrRemoveServicePermissionsRequestBody();
        List<String> listbodyPermissions = new ArrayList<>();
        listbodyPermissions.add("iam:domain::fc973eea581490997e82ea11a1d0101");
        body.withAction(AddOrRemoveServicePermissionsRequestBody.ActionEnum.fromValue("add"));
        body.withPermissions(listbodyPermissions);
        request.withBody(body);
        try {
            AddOrRemoveServicePermissionsResponse response =
client.addOrRemoveServicePermissions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 移除终端节点服务的白名单

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

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

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

public class AddOrRemoveServicePermissionsSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        AddOrRemoveServicePermissionsRequest request = new
AddOrRemoveServicePermissionsRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        AddOrRemoveServicePermissionsRequestBody body = new
```

```
AddOrRemoveServicePermissionsRequestBody();
    List<String> listbodyPermissions = new ArrayList<>();
    listbodyPermissions.add("iam:domain::fc973eea581490997e82ea11a1d0101");

body.withAction(AddOrRemoveServicePermissionsRequestBody.ActionEnum.fromValue("remove"));
body.withPermissions(listbodyPermissions);
request.withBody(body);
try {
    AddOrRemoveServicePermissionsResponse response =
client.addOrRemoveServicePermissions(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- 添加终端节点服务的白名单

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddOrRemoveServicePermissionsRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listPermissionsbody = [
            "iam:domain::fc973eea581490997e82ea11a1d0101"
        ]
        request.body = AddOrRemoveServicePermissionsRequestBody(
            action="add",
            permissions=listPermissionsbody
        )
        response = client.add_or_remove_service_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

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

- 移除终端节点服务的白名单

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddOrRemoveServicePermissionsRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listPermissionsbody = [
            "iam:domain::fc973eea581490997e82ea11a1d0101"
        ]
        request.body = AddOrRemoveServicePermissionsRequestBody(
            action="remove",
            permissions=listPermissionsbody
        )
        response = client.add_or_remove_service_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 添加终端节点服务的白名单

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

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

client := vpcep.NewVpcepClient(
    vpcep.VpcepClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.AddOrRemoveServicePermissionsRequest{}
request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
var listPermissionsbody = []string{
    "iam:domain::fc973eea581490997e82ea11a1d0101",
}
request.Body = &model.AddOrRemoveServicePermissionsRequestBody{
    Action: model.GetAddOrRemoveServicePermissionsRequestBodyActionEnum().ADD,
    Permissions: listPermissionsbody,
}
response, err := client.AddOrRemoveServicePermissions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 移除终端节点服务的白名单

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddOrRemoveServicePermissionsRequest{}
    request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
    var listPermissionsbody = []string{
```

```
    "iam:domain::fc973eea581490997e82ea11a1d0101",
  }
  request.Body = &model.AddOrRemoveServicePermissionsRequestBody{
    Action: model.GetAddOrRemoveServicePermissionsRequestBodyActionEnum().REMOVE,
    Permissions: listPermissionsbody,
  }
  response, err := client.AddOrRemoveServicePermissions(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.10 查询公共终端节点服务列表

功能介绍

查询公共终端节点服务的列表，公共终端节点服务是所有用户可见且可连接的终端节点服务，

由运维人员创建，用户可直接使用，但无权创建。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services/public

表 4-62 路径参数

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

表 4-63 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	查询返回公共的终端节点服务数量限制，即每页返回的个数。 取值范围：0~1000，取值一般为10，20或者50，默认为10。
offset	否	Integer	偏移量。 偏移量为一个大于0小于终端节点服务总个数的整数， 表示从偏移量后面的终端节点服务开始查询。
endpoint_service_name	否	String	公共终端节点服务的名称，支持大小写以及模糊匹配。
id	否	String	公共终端节点服务的ID，唯一标识。
sort_key	否	String	查询结果中终端节点服务列表的排序字段，取值为： <ul style="list-style-type: none">• create_at：终端节点服务的创建时间• update_at：终端节点服务的更新时间 默认值为create_at。
sort_dir	否	String	查询结果中终端节点服务列表的排序方式，取值为： <ul style="list-style-type: none">• desc：降序排序• asc：升序排序 默认值为desc。

请求参数

表 4-64 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)。

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-65 响应 Body 参数

参数	参数类型	描述
endpoint_services	Array of EndpointService objects	终端节点服务列表。
total_count	Integer	满足查询条件的公共终端节点服务总条数，不受分页（即limit、offset参数）影响。

表 4-66 EndpointService

参数	参数类型	描述
id	String	公共终端节点服务的ID，唯一标识。
owner	String	终端节点服务的所有者。
service_name	String	公共终端节点服务的名称。
service_type	String	终端节点服务类型。 <ul style="list-style-type: none">gateway：由运维人员配置。用户无需创建，可直接使用。interface：包括运维人员配置的云服务和用户自己创建的私有服务。其中，运维人员配置的云服务无需创建，用户可直接使用。 您可以通过创建终端节点创建访问Gateway和Interface类型终端节点服务的终端节点。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ

参数	参数类型	描述
is_charge	Boolean	连接该终端节点服务的终端节点是否计费。 <ul style="list-style-type: none">• true: 计费• false: 不计费
enable_policy	Boolean	是否允许自定义终端节点策略。 <ul style="list-style-type: none">• false: 不支持设置终端节点策略• true: 支持设置终端节点策略 默认为false

请求示例

查询公共终端节点服务列表

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/public?
limit={limit}&offset={offset}&endpoint_service_name={endpoint_service_name}&id={endpoint_service_id}&so
rt_key={sort_key}&sort_dir={sort_dir}
```

响应示例

状态码: 200

服务器已成功处理了请求

```
{
  "endpoint_services": [ {
    "id": "b0e22f6f-26f4-461c-b140-d873464d4fa0",
    "owner": "example",
    "service_name": "test123",
    "service_type": "interface",
    "created_at": "2018-09-10T13:13:23Z",
    "is_charge": "true"
  }, {
    "id": "26391a76-546b-42a9-b2fc-496ec68c0e4d",
    "owner": "example",
    "service_name": "OBS",
    "service_type": "gateway",
    "created_at": "2019-03-28T09:30:27Z",
    "is_charge": "true"
  } ],
  "total_count": 2
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
import com.huaweicloud.sdk.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class ListServicePublicDetailsSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        ListServicePublicDetailsRequest request = new ListServicePublicDetailsRequest();
        try {
            ListServicePublicDetailsResponse response = client.listServicePublicDetails(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = VpcepClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(VpcepRegion.value_of("<YOUR REGION>")) \
```

```
.build()

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.11 查询终端节点服务概要

功能介绍

查询终端节点服务的概要信息，此接口是供创建终端节点的用户来查询需要连接的终端节点服务信息。此接口既可以方便其他用户查询到您的终端节点服务概要信息，又可以避免您的终端节点服务的细节信息暴露给其他用户。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoint-services/describe

表 4-67 路径参数

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

表 4-68 Query 参数

参数	是否必选	参数类型	描述
endpoint_service_name	否	String	终端节点服务的名称。说明：该字段和id字段必须二选一，否则会出现错误。
id	否	String	终端节点服务的ID，唯一标识。说明：该字段必须和endpoint_service_name字段二选一，否则会出现错误。

请求参数

表 4-69 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-70 响应 Body 参数

参数	参数类型	描述
id	String	终端节点服务的ID，唯一标识。
service_name	String	终端节点服务的名称。
service_type	String	终端节点服务类型。仅支持将用户私有服务创建为interface类型的终端节点服务。 <ul style="list-style-type: none">gateway：由运维人员配置。用户无需创建，可直接使用。interface：包括运维人员配置的云服务和用户自己创建的私有服务。其中，运维人员配置的云服务无需创建，用户可直接使用。您可以通过创建终端节点创建访问Gateway和Interface类型终端节点服务的终端节点。
created_at	String	终端节点服务的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
is_charge	Boolean	连接该终端节点服务的终端节点是否计费。 <ul style="list-style-type: none">true：计费false：不计费

参数	参数类型	描述
public_border_group	String	终端节点对应Pool的Public Border Group信息
enable_policy	Boolean	是否允许自定义终端节点策略。 <ul style="list-style-type: none">• false: 不支持设置终端节点策略• true: 支持设置终端节点策略 默认为false

请求示例

查询终端节点服务概要

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/describe?id=4189d3c2-8882-4871-a3c2-d380272eed83
```

响应示例

状态码: 200

服务器已成功处理了请求

```
{
  "id": "9d4c1028-1336-4556-9881-b5d807c1b8a8",
  "service_name": "test123",
  "service_type": "interface",
  "created_at": "2018-09-17T07:28:31Z",
  "is_charge": "true"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class ListServiceDescribeDetailsSolution {

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

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

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

VpcepClient client = VpcepClient.newBuilder()
    .withCredential(auth)
    .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
    .build();
ListServiceDescribeDetailsRequest request = new ListServiceDescribeDetailsRequest();
try {
    ListServiceDescribeDetailsResponse response = client.listServiceDescribeDetails(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

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


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.12 升级终端节点服务

功能介绍

升级终端节点服务，使终端节点服务支持创建专业型终端节点实例

该接口仅支持在华东二、中东-利雅得、华东-青岛、非洲-开罗局点调用。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/upgrade

表 4-71 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-72 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)。
Content-Type	是	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

无

请求示例

升级终端节点服务，使终端节点服务支持创建专业型终端节点实例

```
POST https://{endpoint}/v2/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/upgrade

{
  "permissions" : [ "iam:domain::fc973eea581490997e82ea11a1d0101" ],
  "action" : "add"
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

升级终端节点服务，使终端节点服务支持创建专业型终端节点实例

```
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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class UpgradeEndpointServiceSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        UpgradeEndpointServiceRequest request = new UpgradeEndpointServiceRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        try {
            UpgradeEndpointServiceResponse response = client.upgradeEndpointService(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```

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

Python

升级终端节点服务，使终端节点服务支持创建专业型终端节点实例

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpgradeEndpointServiceRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        response = client.upgrade_endpoint_service(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

升级终端节点服务，使终端节点服务支持创建专业型终端节点实例

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

```
projectId := "{project_id}"

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

client := vpcep.NewVpcepClient(
    vpcep.VpcepClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
204	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.13 修改终端节点服务名称

功能介绍

修改终端节点服务名称

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/name

表 4-73 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户ID
vpc_endpoint_service_id	是	String	终端节点服务ID

请求参数

表 4-74 请求 Body 参数

参数	是否必选	参数类型	描述
endpoint_service_name	否	String	终端节点服务名称

响应参数

状态码： 200

表 4-75 响应 Body 参数

参数	参数类型	描述
endpoint_service_name	String	终端节点服务名称

请求示例

修改终端节点服务名称

```
https://{endpoint}/v1/{project_id}/vpc-endpoint-services/070c4a63-8a62-448e-89ed-f2cbde8e6a3b/name  
{  
  "endpoint_service_name": "obs_test"  
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{  
  "endpoint_service_name": "br-abc-aaa1.obs_test.77e40857-03e9-45ee-a940-4bbb8cef72bf"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改终端节点服务名称

```
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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class UpdateEndpointServiceNameSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateEndpointServiceNameRequest request = new UpdateEndpointServiceNameRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        UpdateEndpointServiceNameRequestBody body = new UpdateEndpointServiceNameRequestBody();
        body.withEndpointServiceName("obs_test");
        request.withBody(body);
        try {
            UpdateEndpointServiceNameResponse response = client.updateEndpointServiceName(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改终端节点服务名称

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateEndpointServiceNameRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        request.body = UpdateEndpointServiceNameRequestBody(
            endpoint_service_name="obs_test"
        )
        response = client.update_endpoint_service_name(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

修改终端节点服务名称

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.UpdateEndpointServiceNameRequest{}
request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
endpointServiceNameUpdateEndpointServiceNameRequestBody := "obs_test"
request.Body = &model.UpdateEndpointServiceNameRequestBody{
    EndpointServiceName: &endpointServiceNameUpdateEndpointServiceNameRequestBody,
}
response, err := client.UpdateEndpointServiceName(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.14 更新终端节点连接描述

功能介绍

更新终端节点服务连接的终端节点的描述。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/connections/description

表 4-76 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户ID
vpc_endpoint_service_id	是	String	终端节点服务ID

请求参数

表 4-77 请求 Body 参数

参数	是否必选	参数类型	描述
connections	是	Array of Connections Desc objects	连接管理描述字段列表

表 4-78 ConnectionsDesc

参数	是否必选	参数类型	描述
id	是	String	终端节点ID，UUID格式字符
description	是	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。

响应参数

状态码： 200

表 4-79 响应 Body 参数

参数	参数类型	描述
connections	Array of ConnectionEndpoints objects	连接列表

表 4-80 ConnectionEndpoints

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
marker_id	Integer	终端节点的报文标识。
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
domain_id	String	用户的Domain ID。

参数	参数类型	描述
error	Array of QueryError objects	错误信息。 当终端节点服务状态异常，即“status”的值为“failed”时，会返回该字段。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
description	String	终端节点连接描述。

表 4-81 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

请求示例

更新终端节点连接描述

```
https://{endpoint}/v1/{project_id}/vpc-endpoint-services/813d9300-8473-405b-bfcd-f711117bad65/connections/description
```

```
{
  "connections": [ {
    "id": "0df78897-35b0-4a93-ad17-0bc0585f546b",
    "description": "终端节点节点1, 属于账户[0605767a3300d5762fb7c0186d9e1779]abc_26"
  }, {
    "id": "0d837f75-90d5-4528-bd11-5d974238ebca",
    "description": "终端节点节点2, 属于账户[787699999999]abc_26"
  }
]
```

响应示例

状态码: 200

服务器已成功处理了请求

```
{
  "connections": [ {
    "id": "0d837f75-90d5-4528-bd11-5d974238ebca",
    "status": "accepted",
    "description": "终端节点节点2, 属于账户[787699999999]abc_26",
    "marker_id": "302008767",
    "domain_id": "05b5408a0a80d2b10f06c0184a774460",
  }
]
```

```
"created_at" : "2022-04-14T02:09:08Z",
"updated_at" : "2022-04-14T07:55:40Z"
}, {
  "id" : "0df78897-35b0-4a93-ad17-0bc0585f546b",
  "status" : "accepted",
  "description" : "终端节点节点1, 属于账户[0605767a3300d5762fb7c0186d9e1779]abc_26",
  "marker_id" : 302001833,
  "domain_id" : "05b5408a0a80d2b10f06c0184a774460",
  "created_at" : "2022-04-14T01:59:31Z",
  "updated_at" : "2022-04-14T07:57:39Z"
}]
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新终端节点连接描述

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

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

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

public class UpdateEndpointConnectionsDescSolution {

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

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

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

        UpdateEndpointConnectionsDescRequest request = new UpdateEndpointConnectionsDescRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        UpdateEndpointConnectionsDescRequestBody body = new
        UpdateEndpointConnectionsDescRequestBody();
        List<ConnectionsDesc> listbodyConnections = new ArrayList<>();
        listbodyConnections.add(
            new ConnectionsDesc()
                .withId("0df78897-35b0-4a93-ad17-0bc0585f546b")
                .withDescription("终端节点节点1, 属于账户[0605767a3300d5762fb7c0186d9e1779]abc_26")
        );
        listbodyConnections.add(
```

```
        new ConnectionsDesc()
            .withId("0d837f75-90d5-4528-bd11-5d974238ebca")
            .withDescription("终端节点节点2, 属于账户[787699999999]abc_26")
    );
    body.withConnections(listbodyConnections);
    request.withBody(body);
    try {
        UpdateEndpointConnectionsDescResponse response =
client.updateEndpointConnectionsDesc(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
    }
}
```

Python

更新终端节点连接描述

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateEndpointConnectionsDescRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listConnectionsbody = [
            ConnectionsDesc(
                id="0df78897-35b0-4a93-ad17-0bc0585f546b",
                description="终端节点节点1, 属于账户[0605767a3300d5762fb7c0186d9e1779]abc_26"
            ),
            ConnectionsDesc(
                id="0d837f75-90d5-4528-bd11-5d974238ebca",
                description="终端节点节点2, 属于账户[787699999999]abc_26"
            )
        ]
        request.body = UpdateEndpointConnectionsDescRequestBody(
            connections=listConnectionsbody
        )
```

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

Go

更新终端节点连接描述

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEndpointConnectionsDescRequest{}
    request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
    var listConnectionsbody = []model.ConnectionsDesc{
        {
            Id: "0df78897-35b0-4a93-ad17-0bc0585f546b",
            Description: "终端节点节点1, 属于账户[0605767a3300d5762fb7c0186d9e1779]abc_26",
        },
        {
            Id: "0d837f75-90d5-4528-bd11-5d974238ebca",
            Description: "终端节点节点2, 属于账户[7876999999999]abc_26",
        },
    }
    request.Body = &model.UpdateEndpointConnectionsDescRequestBody{
        Connections: listConnectionsbody,
    }
    response, err := client.UpdateEndpointConnectionsDesc(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.15 批量添加终端节点服务的白名单

功能介绍

批量添加当前用户下终端节点服务的白名单，支持添加描述信息。

说明

本账号默认在自身用户的终端节点服务的白名单中。

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions/batch-create

表 4-82 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-83 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-84 请求 Body 参数

参数	是否必选	参数类型	描述
permissions	是	Array of EpsAddPermissionRequest objects	终端节点服务白名单列表
permission_type	否	String	终端节点服务白名单类型。 <ul style="list-style-type: none">• domainId: 基于账户ID配置终端节点服务白名单。• orgPath: 基于账户所在组织路径配置终端节点服务白名单。

表 4-85 EpsAddPermissionRequest

参数	是否必选	参数类型	描述
permission	是	String	<p>权限格式为： iam:domain::domain_id或者 organizations:orgPath::org_path其中，</p> <ul style="list-style-type: none"> “iam:domain:” 和 “organizations:orgPath:” 为固定格式。 “domain_id” 为可连接用户的账号ID，org_path可连接用户的组织路径 <p>domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“，最大长度可以传64；org_path类型支持“a~z”、“A~Z”、“0~9”、“/”或者“”，最大长度可以传1024；例如： iam:domain::6e9dfd51d1124e8d8498dce894923a0dd或者 organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbf i578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltaws m6qct5z7rklv/</p>
description	是	String	终端节点服务白名单描述

响应参数

状态码： 200

表 4-86 响应 Body 参数

参数	参数类型	描述
permissions	Array of EpsPermission objects	终端节点服务白名单

表 4-87 EpsPermission

参数	参数类型	描述
id	String	白名单表主键ID
permission	String	<p>权限格式为：iam:domain::domain_id或者organizations:orgPath::org_path其中，</p> <ul style="list-style-type: none"> “iam:domain::”和“organizations:orgPath::”为固定格式。 “domain_id”为可连接用户的账号ID，org_path可连接用户的组织路径domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“”，org_path类型支持“a~z”、“A~Z”、“0~9”、“/”或者“”。 “”表示所有终端节点可连接。 <p>例如： iam:domain::6e9dfd51d1124e8d8498dce894923a0dd或者 organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnd5rgag77m2z/ ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*</p>
permission_type	String	<p>终端节点服务白名单类型。</p> <ul style="list-style-type: none"> domainId：基于账户ID配置终端节点服务白名单。 orgPath：基于账户所在组织路径配置终端节点服务白名单。
description	String	终端节点服务白名单描述
created_at	String	白名单创建时间

请求示例

批量添加终端节点服务的白名单

POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/c07132bf-4071-439a-9ff3-2068eb45092f/permissions/batch-create

```
{
  "permissions": [ {
    "permission": "*",
    "description": "批量添加终端节点服务白名单接口"
  }, {
    "permission": "iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",
    "description": "test"
  }
]
```

```
    }  
  }  
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{  
  "permissions": [{  
    "id": "ab42a58b-6f8e-43b3-9400-aaf0e393ee0d",  
    "permission": "iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",  
    "description": "test",  
    "created_at": "2022-06-16T10:03:27Z"  
  }, {  
    "id": "dfaeff1-858d-4631-83c4-be3548a07935",  
    "permission": "*",  
    "description": "",  
    "created_at": "2022-06-16T10:03:27Z"  
  }  
}]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量添加终端节点服务的白名单

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.vpcep.v1.region.VpcepRegion;  
import com.huaweicloud.sdk.vpcep.v1.*;  
import com.huaweicloud.sdk.vpcep.v1.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class BatchAddEndpointServicePermissionsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        VpcepClient client = VpcepClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))  
            .build();
```

```
BatchAddEndpointServicePermissionsRequest request = new
BatchAddEndpointServicePermissionsRequest();
request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
BatchAddEndpointServicePermissionsRequestBody body = new
BatchAddEndpointServicePermissionsRequestBody();
List<EpsAddPermissionRequest> listbodyPermissions = new ArrayList<>();
listbodyPermissions.add(
    new EpsAddPermissionRequest()
        .withPermission("iam")
        .withDescription("批量添加终端节点服务白名单接口")
);
listbodyPermissions.add(
    new EpsAddPermissionRequest()
        .withPermission("iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397")
        .withDescription("test")
);
body.withPermissions(listbodyPermissions);
request.withBody(body);
try {
    BatchAddEndpointServicePermissionsResponse response =
client.batchAddEndpointServicePermissions(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

批量添加终端节点服务的白名单

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchAddEndpointServicePermissionsRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listPermissionsbody = [
```

```
EpsAddPermissionRequest(  
    permission="*",  
    description="批量添加终端节点服务白名单接口"  
),  
EpsAddPermissionRequest(  
    permission="iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",  
    description="test"  
)  
]  
request.body = BatchAddEndpointServicePermissionsRequestBody(  
    permissions=listPermissionsbody  
)  
response = client.batch_add_endpoint_service_permissions(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

批量添加终端节点服务的白名单

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := vpcep.NewVpcepClient(  
        vpcep.VpcepClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.BatchAddEndpointServicePermissionsRequest{}  
    request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"  
    var listPermissionsbody = []model.EpsAddPermissionRequest{  
        {  
            Permission: "*",  
            Description: "批量添加终端节点服务白名单接口",  
        },  
        {  
            Permission: "iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",  
            Description: "test",  
        },  
    }  
    request.Body = &model.BatchAddEndpointServicePermissionsRequestBody{
```

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.16 批量删除终端节点服务的白名单

功能介绍

批量删除当前用户下终端节点服务的白名单

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions/batch-delete

表 4-88 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。

请求参数

表 4-89 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-90 请求 Body 参数

参数	是否必选	参数类型	描述
permissions	是	Array of EpsRemovePermissionRequest objects	终端节点服务白名单

表 4-91 EpsRemovePermissionRequest

参数	是否必选	参数类型	描述
id	是	String	终端节点服务白名单表主键ID

响应参数

状态码： 200

表 4-92 响应 Body 参数

参数	参数类型	描述
permissions	Array of EpsPermission objects	终端节点服务白名单

表 4-93 EpsPermission

参数	参数类型	描述
id	String	白名单表主键ID
permission	String	<p>权限格式为：iam:domain::domain_id或者organizations:orgPath::org_path其中，</p> <ul style="list-style-type: none"> “iam:domain::”和“organizations:orgPath::”为固定格式。 “domain_id”为可连接用户的账号ID，org_path可连接用户的组织路径domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“”，org_path类型支持“a~z”、“A~Z”、“0~9”、“/”或者“”。 “”表示所有终端节点可连接。 <p>例如： iam:domain::6e9dfd51d1124e8d8498dce894923a0dd或者 organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rlbdu1vmxdw5ahdkknxnd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*</p>
permission_type	String	<p>终端节点服务白名单类型。</p> <ul style="list-style-type: none"> domainId：基于账户ID配置终端节点服务白名单。 orgPath：基于账户所在组织路径配置终端节点服务白名单。
description	String	终端节点服务白名单描述
created_at	String	白名单创建时间

请求示例

批量删除终端节点服务的白名单

POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/c07132bf-4071-439a-9ff3-2068eb45092f/permissions/batch-delete

```
{
  "permissions": [ {
    "id": "b9b6f264-3562-4f62-ba84-c128da20d18d"
  }, {
    "id": "c07132bf-4071-439a-9ff3-2068eb45092f"
  } ]
}
```


响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "permissions" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

批量删除终端节点服务的白名单

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

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

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

public class BatchRemoveEndpointServicePermissionsSolution {

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

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

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

        BatchRemoveEndpointServicePermissionsRequest request = new
        BatchRemoveEndpointServicePermissionsRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        BatchRemoveEndpointServicePermissionsRequestBody body = new
        BatchRemoveEndpointServicePermissionsRequestBody();
        List<EpsRemovePermissionRequest> listbodyPermissions = new ArrayList<>();
        listbodyPermissions.add(
            new EpsRemovePermissionRequest()
                .withId("b9b6f264-3562-4f62-ba84-c128da20d18d")
        );
        listbodyPermissions.add(
            new EpsRemovePermissionRequest()
                .withId("c07132bf-4071-439a-9ff3-2068eb45092f")
        );
    }
}
```

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

Python

批量删除终端节点服务的白名单

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchRemoveEndpointServicePermissionsRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        listPermissionsbody = [
            EpsRemovePermissionRequest(
                id="b9b6f264-3562-4f62-ba84-c128da20d18d"
            ),
            EpsRemovePermissionRequest(
                id="c07132bf-4071-439a-9ff3-2068eb45092f"
            )
        ]
        request.body = BatchRemoveEndpointServicePermissionsRequestBody(
            permissions=listPermissionsbody
        )
        response = client.batch_remove_endpoint_service_permissions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

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

Go

批量删除终端节点服务的白名单

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRemoveEndpointServicePermissionsRequest{}
    request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
    var listPermissionsbody = []model.EpsRemovePermissionRequest{
        {
            Id: "b9b6f264-3562-4f62-ba84-c128da20d18d",
        },
        {
            Id: "c07132bf-4071-439a-9ff3-2068eb45092f",
        },
    }
    request.Body = &model.BatchRemoveEndpointServicePermissionsRequestBody{
        Permissions: listPermissionsbody,
    }
    response, err := client.BatchRemoveEndpointServicePermissions(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.2.17 更新终端节点服务白名单描述

功能介绍

更新当前用户下终端节点服务白名单的描述信息

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions/{permission_id}

表 4-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_service_id	是	String	终端节点服务的ID。
permission_id	是	String	白名单的ID。

请求参数

表 4-95 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)。

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-96 请求 Body 参数

参数	是否必选	参数类型	描述
permission	是	EpsUpdatePermissionDesc object	更新终端节点服务白名单描述的请求体

表 4-97 EpsUpdatePermissionDesc

参数	是否必选	参数类型	描述
description	是	String	终端节点服务白名单描述

响应参数

状态码： 200

表 4-98 响应 Body 参数

参数	参数类型	描述
permissions	Array of EpsPermission objects	终端节点服务白名单

表 4-99 EpsPermission

参数	参数类型	描述
id	String	白名单表主键ID

参数	参数类型	描述
permission	String	权限格式为：iam:domain::domain_id或者organizations:orgPath::org_path其中， <ul style="list-style-type: none">“iam:domain::”和“organizations:orgPath::”为固定格式。“domain_id”为可连接用户的账号ID，org_path可连接用户的组织路径domain_id类型支持输入包括“a~z”、“A~Z”、“0~9”或者“”，org_path类型支持“a~z”、“A~Z”、“0~9”、“/~-?”或者“”。 “”表示所有终端节点可连接。 例如： iam:domain::6e9dfd51d1124e8d8498dce894923a0dd或者 organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnv55rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*
permission_type	String	终端节点服务白名单类型。 <ul style="list-style-type: none">domainId：基于账户ID配置终端节点服务白名单。orgPath：基于账户所在组织路径配置终端节点服务白名单。
description	String	终端节点服务白名单描述
created_at	String	白名单创建时间

请求示例

更新终端节点服务白名单描述

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoint-services/c07132bf-4071-439a-9ff3-2068eb45092f/permissions/849046d9-7342-44ba-b2d2-367d8de6311b
```

```
{
  "permission": {
    "description": "终端节点服务添加白名单"
  }
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "permissions": [ {
    "id": "849046d9-7342-44ba-b2d2-367d8de6311b",
    "permission": "*",
    "description": "终端节点服务添加白名单",
    "created_at": "2022-06-16T09:48:21Z"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新终端节点服务白名单描述

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

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

public class UpdateEndpointServicePermissionDescSolution {

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

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

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

        UpdateEndpointServicePermissionDescRequest request = new
        UpdateEndpointServicePermissionDescRequest();
        request.withVpcEndpointServiceId("{vpc_endpoint_service_id}");
        request.withPermissionId("{permission_id}");
        UpdateEndpointServicePermissionDescRequestBody body = new
        UpdateEndpointServicePermissionDescRequestBody();
        EpsUpdatePermissionDesc permissionbody = new EpsUpdatePermissionDesc();
        permissionbody.withDescription("终端节点服务添加白名单");
        body.withPermission(permissionbody);
        request.withBody(body);
        try {
            UpdateEndpointServicePermissionDescResponse response =
            client.updateEndpointServicePermissionDesc(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

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

Python

更新终端节点服务白名单描述

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateEndpointServicePermissionDescRequest()
        request.vpc_endpoint_service_id = "{vpc_endpoint_service_id}"
        request.permission_id = "{permission_id}"
        permissionbody = EpsUpdatePermissionDesc(
            description="终端节点服务添加白名单"
        )
        request.body = UpdateEndpointServicePermissionDescRequestBody(
            permission=permissionbody
        )
        response = client.update_endpoint_service_permission_desc(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新终端节点服务白名单描述

```
package main

import (
    "fmt"
```



```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEndpointServicePermissionDescRequest{}
    request.VpcEndpointServiceId = "{vpc_endpoint_service_id}"
    request.PermissionId = "{permission_id}"
    permissionbody := &model.EpsUpdatePermissionDesc{
        Description: "终端节点服务添加白名单",
    }
    request.Body = &model.UpdateEndpointServicePermissionDescRequestBody{
        Permission: permissionbody,
    }
    response, err := client.UpdateEndpointServicePermissionDesc(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3 终端节点功能

4.3.1 创建终端节点

功能介绍

创建终端节点，以便访问终端节点服务。

该接口为异步接口，调用成功会返回200状态码，说明请求已正常下发。

通常创建终端节点需要1~2分钟，可以通过[查询终端节点详情](#)查看创建结果

调用方法

请参见[如何调用API](#)。

URI

POST /v1/{project_id}/vpc-endpoints

表 4-100 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 4-101 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-102 请求 Body 参数

参数	是否必选	参数类型	描述
subnet_id	否	String	创建连接Interface类型终端节点服务的终端节点时，此参数必选。需要指定vpc_id对应VPC下已创建的网络（network）的ID，UUID格式。说明：- VPC的子网网段不能与198.19.128.0/17重叠 - VPC路由表中自定义路由的目的地址不能与198.19.128.0/17重叠
endpoint_service_id	是	String	终端节点服务的ID。 可以通过查询终端节点服务概要获取要连接的终端节点服务ID。
vpc_id	是	String	终端节点所在的VPC的ID。
enable_dns	否	Boolean	是否创建域名。 <ul style="list-style-type: none">• true: 创建域名• false: 不创建域名 默认值为false。 当创建gateway类型终端节点服务的终端节点时，“enable_dns”设置为true或者false，均不创建域名。
tags	否	Array of TagList objects	标签列表，没有标签默认为空数组。
routetables	否	Array of strings	路由表ID列表。 创建gateway类型终端节点服务的终端节点时，此参数必选。 不设置此参数时，选择默认路由表。
port_ip	否	String	访问所连接的终端节点服务的IP。 创建终端节点时，可以指定访问所连接的终端节点服务的IP，目前只支持IPv4类型。 创建连接Interface类型终端节点服务的终端节点时，此参数必选。

参数	是否必选	参数类型	描述
whitelist	否	Array of strings	添加用于控制访问终端节点的白名单。 创建终端节点时，支持访问控制，使用此参数可以添加IPv4或CIDR，默认空列表。 仅当创建连接Interface类型终端节点服务的终端节点时，支持设置此参数。
enable_whitelist	否	Boolean	是否开启网络ACL隔离。
description	否	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
policy_statement	否	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。
policy_document	否	Object	终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）
ip_version	否	String	指定终端节点的IP版本，仅专业型终端节点支持此参数。 <ul style="list-style-type: none">• ipv4, IPv4• dualstack, 双栈
ipv6_address	否	String	访问所连接的终端节点服务的IPv6的地址。 创建终端节点时，可以指定访问所连接的终端节点服务的IP，不指定的情况下，会使用系统生成的一个地址。 仅专业型终端节点支持此参数。

表 4-103 TagList

参数	是否必选	参数类型	描述
key	否	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符： _ . : = + - @。 _sys_ 开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	否	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符： _ . : / = + - @。 资源标签值可以为空字符串。

表 4-104 PolicyStatement

参数	是否必选	参数类型	描述
Effect	是	String	<ul style="list-style-type: none">• Allow, 允许控制访问权限• Deny, 拒绝控制访问权限
Action	是	Array of strings	obs访问权限
Resource	是	Array of strings	obs对象

响应参数

状态码： 200

表 4-105 响应 Body 参数

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。

参数	参数类型	描述
service_type	String	终端节点连接的终端节点服务类型。 <ul style="list-style-type: none">gateway: 由运维人员配置。用户无需创建, 可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中, 运维人员配置的云服务无需创建, 用户可直接使用。 您可以通过"查询公共终端节点服务列表"查看由运维人员配置的所有用户可见且可连接的终端节点服务, 并通过创建终端节点服务创建Interface类型的终端节点服务。
status	String	终端节点的状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
ip	String	终端节点ip
active_status	Array of strings	账号状态。 <ul style="list-style-type: none">frozen: 冻结active: 解冻
endpoint_service_name	String	终端节点服务的名称。
marker_id	Integer	终端节点的报文标识。
endpoint_service_id	String	终端节点服务的ID。
enable_dns	Boolean	是否创建域名。 <ul style="list-style-type: none">true: 创建域名false: 不创建域名 当创建gateway类型终端节点服务的终端节点时, "enable_dns" 设置为true或者false, 均不创建域名。
subnet_id	String	vpc_id对应VPC下已创建的网络(network)的ID, UUID格式。
vpc_id	String	终端节点所在的VPC的ID。

参数	参数类型	描述
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
tags	Array of TagList objects	标签列表，没有标签默认为空数组。
whitelist	Array of strings	控制访问终端节点的白名单。 若未创建，则返回空列表。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
enable_whitelist	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">• true：开启网络ACL隔离• false：不开启网络ACL隔离 若未指定，则返回false。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
routetables	Array of strings	路由表ID列表。 若未指定，返回默认VPC下路由表ID。 创建gateway类型终端节点服务的终端节点时，显示此参数。
specification_name	String	规格名称
description	String	描述
policy_statement	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。
policy_document	Object	终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）
enable_status	String	终端节点是否可用。 <ul style="list-style-type: none">• enable：启用• disable：不启用
endpoint_pool_id	String	待废弃，实例相关联的集群ID

参数	参数类型	描述
public_border_group	String	终端节点对应Pool的Public Border Group信息
ipv6_address	String	终端节点的IPv6地址,仅专业型终端节点支持此参数

表 4-106 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-107 PolicyStatement

参数	参数类型	描述
Effect	String	<ul style="list-style-type: none"> Allow, 允许控制访问权限 Deny, 拒绝控制访问权限
Action	Array of strings	obs访问权限
Resource	Array of strings	obs对象

请求示例

创建一个终端节点，不创建域名。

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoints
{
```



```
"subnet_id": "5d1c1d71-2613-4274-b34e-d82af550f967",
"vpc_id": "4e65f8b1-306d-4522-8ecd-aa374000e2a4",
"endpoint_service_id": "813d9300-8473-405b-bfcd-f711117bad65",
"enable_dns": "false"
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "id": "492c776f-2cb9-40a2-9735-d7279ab2bbb6",
  "status": "accepted",
  "ip": "172.16.0.154",
  "tags": [ ],
  "whitelist": [ ],
  "marker_id": "302010104",
  "active_status": [ "active" ],
  "vpc_id": "0da03835-1dcf-4361-9b87-34139d58dd59",
  "service_type": "interface",
  "project_id": "0605767a3300d5762fb7c0186d9e1779",
  "subnet_id": "fb1b8b7e-f34c-4ff1-b3a7-221acef3b3aa",
  "enable_dns": false,
  "created_at": "2022-04-28T02:44:50Z",
  "updated_at": "2022-04-28T02:56:15Z",
  "endpoint_service_id": "8843fac2-bc01-445a-a822-c8124ff3968c",
  "endpoint_service_name": "br-abc-aaa1.vm_test.8843fac2-bc01-445a-a822-c8124ff3968c",
  "enable_whitelist": false,
  "specification_name": "default",
  "endpoint_pool_id": "501f4a3b-6f96-4309-97d1-e291b8ca5b96",
  "enable_status": "enable",
  "public_border_group": "center"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个终端节点，不创建域名。

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

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

public class CreateEndpointSolution {

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

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

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

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

CreateEndpointRequest request = new CreateEndpointRequest();
CreateEndpointRequestBody body = new CreateEndpointRequestBody();
body.withEnableDns(false);
body.withVpclid("4e65f8b1-306d-4522-8ecd-aa374000e2a4");
body.withEndpointServiceId("813d9300-8473-405b-bfcd-f711117bad65");
body.withSubnetId("5d1c1d71-2613-4274-b34e-d82af550f967");
request.withBody(body);
try {
    CreateEndpointResponse response = client.createEndpoint(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建一个终端节点，不创建域名。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateEndpointRequest()
        request.body = CreateEndpointRequestBody(
            enable_dns=False,
            vpclid="4e65f8b1-306d-4522-8ecd-aa374000e2a4",
```

```
        endpoint_service_id="813d9300-8473-405b-bfcd-f711117bad65",
        subnet_id="5d1c1d71-2613-4274-b34e-d82af550f967"
    )
    response = client.create_endpoint(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

创建一个终端节点，不创建域名。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateEndpointRequest{
        enableDnsCreateEndpointRequestBody:= false
        subnetIdCreateEndpointRequestBody:= "5d1c1d71-2613-4274-b34e-d82af550f967"
        request.Body = &model.CreateEndpointRequestBody{
            EnableDns: &enableDnsCreateEndpointRequestBody,
            VpcId: "4e65f8b1-306d-4522-8ecd-aa374000e2a4",
            EndpointServiceId: "813d9300-8473-405b-bfcd-f711117bad65",
            SubnetId: &subnetIdCreateEndpointRequestBody,
        }
    }
    response, err := client.CreateEndpoint(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.2 查询终端节点列表

功能介绍

查询当前用户下的终端节点的列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoints

表 4-108 路径参数

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

表 4-109 Query 参数

参数	是否必选	参数类型	描述
endpoint_service_name	否	String	终端节点服务的名称，支持大小写，前后模糊匹配。
vpc_id	否	String	终端节点所在的VPC的ID。
id	否	String	终端节点的ID，唯一标识。
limit	否	Integer	查询返回终端节点的数量限制，即每页返回的资源个数。 取值范围：0~1000，取值一般为10，20或者50，默认为10。

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量。 偏移量为一个大于0小于终端节点服务总个数的整数， 表示从偏移量后面的终端节点服务开始查询。
sort_key	否	String	查询结果中终端节点列表的排序字段，取值为： <ul style="list-style-type: none">• create_at: 终端节点的创建时间• update_at: 终端节点的更新时间 默认值为create_at。
sort_dir	否	String	查询结果中终端节点列表的排序方式，取值为： <ul style="list-style-type: none">• desc: 降序排序• asc: 升序排序 默认值为desc。

请求参数

表 4-110 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-111 响应 Body 参数

参数	参数类型	描述
endpoints	Array of EndpointResponseBody objects	终端节点列表。
total_count	Integer	满足查询条件的终端节点总条数，不受分页（即limit、offset参数）影响。

表 4-112 EndpointResponseBody

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
service_type	String	终端节点连接的终端节点服务类型。 <ul style="list-style-type: none">gateway: 由运维人员配置。用户无需创建，可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中，运维人员配置的云服务无需创建，用户可直接使用。您可以通过查询公共终端节点服务列表，查看由运维人员配置的所有用户可见且可连接的终端节点服务，并通过创建终端节点服务创建Interface类型的终端节点服务。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
active_status	Array of strings	账号状态。 <ul style="list-style-type: none">frozen: 冻结active: 解冻
endpoint_service_name	String	终端节点服务的名称。
marker_id	Integer	终端节点的报文标识。

参数	参数类型	描述
endpoint_service_id	String	终端节点服务的ID。
enable_dns	Boolean	是否创建域名。 <ul style="list-style-type: none">• true: 创建域名• false: 不创建域名 当创建gateway类型终端节点服务的终端节点时，“enable_dns”设置为true或者false，均不创建域名。
dns_names	Array of strings	访问所连接的终端节点服务的域名。当“enable_dns”为true时，该参数可见。
ip	String	访问所连接的终端节点服务的IP。仅当同时满足如下条件时，返回该参数： 当查询连接interface类型终端节点服务的终端节点时。 终端节点服务启用“连接审批”功能，且已经“接受”连接审批。 “status”可以是“accepted”或者“rejected（仅支持“接受”连接审批后再“拒绝”的情况）”。
vpc_id	String	终端节点所在的VPC的ID。
subnet_id	String	vpc_id对应VPC下已创建的网络（network）的ID，UUID格式。
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
tags	Array of TagList objects	标签列表，没有标签默认为空数组。
error	Array of QueryError objects	错误信息。 当终端节点状态异常，即“status”的值为“failed”时，会返回该字段。

参数	参数类型	描述
whitelist	Array of strings	控制访问终端节点的白名单。 若未创建，则返回空列表。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
enable_whitelist	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">• true: 开启网络ACL隔离• false: 不开启网络ACL隔离 若未指定，则返回false。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
routetables	Array of strings	路由表ID列表。 若未指定，返回默认VPC下路由表ID。 创建gateway类型终端节点服务的终端节点时，显示此参数。
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。
policy_statement	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。
policy_document	Object	终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）
endpoint_pool_id	String	待废弃，实例相关联的集群ID
public_border_group	String	终端节点关联的Public Border Group信息，只有当终端节点和边缘Pool相关联时才会返回该字段

表 4-113 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-114 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

表 4-115 PolicyStatement

参数	参数类型	描述
Effect	String	<ul style="list-style-type: none">Allow，允许控制访问权限Deny，拒绝控制访问权限
Action	Array of strings	obs访问权限
Resource	Array of strings	obs对象

请求示例

查询终端节点列表

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoints?  
endpoint_service_name={endpoint_service_name}&vpc_id={vpc_id}&limit={limit}&offset={offset}&id={id}&sort_key={sort_key}&sort_dir={sort_dir}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "endpoints": [ {
    "id": "03184a04-95d5-4555-86c4-e767a371ff99",
    "status": "accepted",
    "ip": "192.168.0.232",
    "marker_id": 16777337,
    "active_status": "active",
    "vpc_id": "84758cf5-9c62-43ae-a778-3dbd8370c0a4",
    "service_type": "interface",
    "project_id": "295dacf46a4842fcbf7844dc2dc2489d",
    "subnet_id": "68bfbcc1-dff2-47e4-a9d4-332b9bc1b8de",
    "enable_dns": "true",
    "dns_names": [ "test123" ],
    "created_at": "2018-10-18T06:49:46Z",
    "updated_at": "2018-10-18T06:49:50Z",
    "endpoint_service_id": "5133655d-0e28-4090-b669-13f87b355c78",
    "endpoint_service_name": "test123",
    "endpoint_pool_id": "ee38223b-aacb-46f0-ba7e-94fa62e35dde",
    "public_border_group": "br-abc-aaa1",
    "whitelist": [ "127.0.0.1" ],
    "enable_whitelist": true
  }, {
    "id": "43b0e3b0-eec9-49da-866b-6687b75f9fe5",
    "status": "accepted",
    "ip": "192.168.0.115",
    "marker_id": 16777322,
    "active_status": "active",
    "vpc_id": "e251b400-2963-4131-b38a-da81e32026ee",
    "service_type": "interface",
    "project_id": "295dacf46a4842fcbf7844dc2dc2489d",
    "subnet_id": "65528a22-59a1-4972-ba64-88984b3207cd",
    "enable_dns": "true",
    "dns_names": [ "test123" ],
    "created_at": "2018-10-18T06:36:20Z",
    "updated_at": "2018-10-18T06:36:24Z",
    "endpoint_service_id": "5133655d-0e28-4090-b669-13f87b355c78",
    "endpoint_service_name": "test123",
    "endpoint_pool_id": "ee38223b-aacb-46f0-ba7e-94fa62e35dde",
    "whitelist": [ "127.0.0.1" ],
    "enable_whitelist": true
  } ],
  "total_count": 2
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

```
public class ListEndpointsSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEndpointsRequest request = new ListEndpointsRequest();
        try {
            ListEndpointsResponse response = client.listEndpoints(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListEndpointsRequest()
```

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.3 查询终端节点详情

功能介绍

查询终端节点的详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v1/{project_id}/vpc-endpoints/{vpc_endpoint_id}

表 4-116 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_id	是	String	终端节点的ID。

请求参数

表 4-117 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

状态码： 200

表 4-118 响应 Body 参数

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
service_type	String	终端节点连接的终端节点服务类型。 <ul style="list-style-type: none">gateway: 由运维人员配置。用户无需创建，可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中，运维人员配置的云服务无需创建，用户可直接使用。您可以通过查询公共终端节点服务列表，查看由运维人员配置的所有用户可见且可连接的终端节点服务，并通过创建终端节点服务创建 Interface 类型的终端节点服务。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
active_status	Array of strings	账号状态。 <ul style="list-style-type: none">frozen: 冻结active: 解冻
enable_status	String	终端节点是否可用。 <ul style="list-style-type: none">enable: 启用disable: 不启用
specification_name	String	终端节点服务规格的名称。
endpoint_service_name	String	终端节点服务的名称。
marker_id	Integer	终端节点的报文标识。
endpoint_service_id	String	终端节点服务的ID。

参数	参数类型	描述
enable_dns	Boolean	是否创建域名。 <ul style="list-style-type: none">• true: 创建域名• false: 不创建域名 当创建gateway类型终端节点服务的终端节点时，“enable_dns”设置为true或者false，均不创建域名。
dns_names	Array of strings	访问所连接的终端节点服务的域名。当“enable_dns”为true时，该参数可见。
ip	String	访问所连接的终端节点服务的IP。仅当同时满足如下条件时，返回该参数： <ul style="list-style-type: none">• 当查询连接interface类型终端节点服务的终端节点时。• 终端节点服务启用“连接审批”功能，且已经“接受”连接审批。“status”可以是“accepted”或者“rejected（仅支持“接受”连接审批后再“拒绝”的情况）”。
vpc_id	String	终端节点所在的VPC的ID。
subnet_id	String	vpc_id对应VPC下已创建的网络（network）的ID，UUID格式。
created_at	String	终端节点的创建时间。采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
tags	Array of TagList objects	标签列表，没有标签默认为空数组。
error	QueryError object	查询资源返回的异常信息
whitelist	Array of strings	控制访问终端节点的白名单。若未创建，则返回空列表。创建连接Interface类型终端节点服务的终端节点时，显示此参数。

参数	参数类型	描述
enable_whitelist	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">• true: 开启网络ACL隔离• false: 不开启网络ACL隔离 若未指定, 则返回false。 创建连接Interface类型终端节点服务的终端节点时, 显示此参数。
routetables	Array of strings	路由表ID列表。 若未指定, 返回默认VPC下路由表ID。 创建gateway类型终端节点服务的终端节点时, 显示此参数。
description	String	描述字段, 支持中英文字母、数字等字符, 不支持“<”或“>”字符。
policy_statement	Array of PolicyStatement objects	Gateway类型终端节点策略信息, 仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。
policy_document	Object	终端节点策略信息, 仅当终端节点服务的enable_policy值为true时支持该参数, 默认值为完全访问权限。(OBS、SFS的终端节点服务暂不支持该参数)
endpoint_pool_id	String	待废弃, 实例相关联的集群ID
public_border_group	String	终端节点对应Pool的Public Border Group信息
ipv6_address	String	访问所连接的终端节点服务的IPv6的地址。 创建终端节点时, 可以指定访问所连接的终端节点服务的IP, 不指定的情况下, 会使用系统生成的一个地址。 仅专业型终端节点支持此参数。

表 4-119 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-120 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

表 4-121 PolicyStatement

参数	参数类型	描述
Effect	String	<ul style="list-style-type: none">Allow, 允许控制访问权限Deny, 拒绝控制访问权限
Action	Array of strings	obs访问权限
Resource	Array of strings	obs对象

请求示例

查询终端节点详情

```
GET https://{endpoint}/v1/{project_id}/vpc-endpoints/4189d3c2-8882-4871-a3c2-d380272eed83
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
  "ip": "192.168.0.187",
  "service_type": "interface",
  "marker_id": "16777337",
  "status": "accepted",
  "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed83",
  "enable_dns": false,
  "endpoint_service_name": "test123",
  "endpoint_service_id": "test123",
  "project_id": "6e9dfd51d1124e8d8498dce894923a0d",
  "whitelist": [ "127.0.0.1" ],
  "enable_whitelist": true,
  "created_at": "2022-04-14T09:35:47Z",
  "updated_at": "2022-04-14T09:36:47Z"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

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

public class ListEndpointInfoDetailsSolution {

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

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

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

        ListEndpointInfoDetailsRequest request = new ListEndpointInfoDetailsRequest();
        request.withVpcEndpointId("{vpc_endpoint_id}");
        try {
            ListEndpointInfoDetailsResponse response = client.listEndpointInfoDetails(request);
            System.out.println(response.toString());
        }
    }
}
```

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

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListEndpointInfoDetailsRequest()
        request.vpc_endpoint_id = "{vpc_endpoint_id}"
        response = client.list_endpoint_info_details(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

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

client := vpcep.NewVpcepClient(
    vpcep.VpcepClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.4 删除终端节点

功能介绍

删除终端节点。

该接口为异步接口，调用成功会返回200状态码，说明请求已正常下发。

通常删除终端节点需要1~2分钟，可以通过[查询终端节点详情](#)查看删除结果

调用方法

请参见[如何调用API](#)。

URI

DELETE /v1/{project_id}/vpc-endpoints/{vpc_endpoint_id}

表 4-122 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。
vpc_endpoint_id	是	String	终端节点的ID。

请求参数

表 4-123 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

响应参数

无

请求示例

删除终端节点

```
DELETE
https://{endpoint}/v1/{project_id}/vpc-endpoints/4189d3c2-8882-4871-a3c2-d380272eed83
```

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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

public class DeleteEndpointSolution {

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

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

        VpcepClient client = VpcepClient.newBuilder()
            .withCredential(auth)
            .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteEndpointRequest request = new DeleteEndpointRequest();
        request.withVpcEndpointId("{vpc_endpoint_id}");
        try {
            DeleteEndpointResponse response = client.deleteEndpoint(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

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

credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = DeleteEndpointRequest()
    request.vpc_endpoint_id = "{vpc_endpoint_id}"
    response = client.delete_endpoint(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

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

状态码

状态码	描述
204	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.5 更新终端节点

功能介绍

更新或删除允许访问终端节点的白名单。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoints/{vpc_endpoint_id}

表 4-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，获取方法请参见获取项目ID。
vpc_endpoint_id	是	String	终端节点的ID。

请求参数

表 4-125 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)。

参数	是否必选	参数类型	描述
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-126 请求 Body 参数

参数	是否必选	参数类型	描述
whitelist	否	Array of strings	更新或删除用于控制访问终端节点的白名单。此参数可以添加IPv4或CIDR： <ul style="list-style-type: none">当取值不为空时，表示将白名单更新为取值所示内容。当取值为空时，表示删除所有白名单。 默认为空列表。
enable_whitelist	否	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">true：开启网络ACL隔离false：不开启网络ACL隔离 默认值为false。

响应参数

状态码： 200

表 4-127 响应 Body 参数

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。

参数	参数类型	描述
service_type	String	终端节点连接的终端节点服务类型。 <ul style="list-style-type: none">gateway: 由运维人员配置, 用户无需创建, 可直接使用。interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中, 运维人员配置的云服务无需创建, 用户可直接使用。您可以通过查询公共终端节点服务列表, 查看由运维人员配置的所有用户可见且可连接的终端节点服务, 并通过创建终端节点服务创建 Interface 类型的终端节点服务。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">pendingAcceptance: 待接受creating: 创建中accepted: 已接受rejected: 已拒绝failed: 失败deleting: 删除中
ip	String	访问所连接的终端节点服务的IP。 仅当同时满足如下条件时, 返回该参数: <ul style="list-style-type: none">当查询连接interface类型终端节点服务的终端节点时。终端节点服务启用“连接审批”功能, 且已经“接受”连接审批。“status”可以是“accepted”或者“rejected (仅支持“接受”连接审批后再“拒绝”的情况)”。
active_status	Array of strings	账号状态。 <ul style="list-style-type: none">frozen: 冻结active: 解冻
endpoint_service_name	String	终端节点服务的名称。
marker_id	Integer	终端节点的报文标识。
endpoint_service_id	String	终端节点服务的ID。

参数	参数类型	描述
enable_dns	Boolean	是否创建域名。 <ul style="list-style-type: none">• true: 创建域名• false: 不创建域名 当创建gateway类型终端节点服务的终端节点时，“enable_dns”设置为true或者false，均不创建域名。
dns_names	Array of strings	访问所连接的终端节点服务的域名。当“enable_dns”为true时，该参数可见。
subnet_id	String	vpc_id对应VPC下已创建的网络（network）的ID，UUID格式。
vpc_id	String	终端节点所在的VPC的ID。
created_at	String	终端节点的创建时间。采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
tags	Array of TagList objects	标签列表，没有标签默认为空数组。
whitelist	Array of strings	控制访问终端节点的白名单。若未创建，则返回空列表。创建连接Interface类型终端节点服务的终端节点时，显示此参数。
enable_whitelist	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">• true: 开启网络ACL隔离• false: 不开启网络ACL隔离 若未指定，则返回false。创建连接Interface类型终端节点服务的终端节点时，显示此参数。
policy_statement	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。

参数	参数类型	描述
policy_document	Object	终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）

表 4-128 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-129 PolicyStatement

参数	参数类型	描述
Effect	String	<ul style="list-style-type: none">Allow, 允许控制访问权限Deny, 拒绝控制访问权限
Action	Array of strings	obs访问权限
Resource	Array of strings	obs对象

请求示例

更新终端节点，设置白名单信息为192.168.1.1、192.168.1.2，开启网络ACL隔离。

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoints/4189d3c2-8882-4871-a3c2-d380272eed83
```

```
{  
  "whitelist":
```

```
[
  "192.168.1.1",
  "192.168.1.2"
],
"enable_whitelist":true,
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
  "service_type": "interface",
  "status": "accepted",
  "ip": "10.32.xx.xx",
  "marker_id": "322312312312",
  "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed83",
  "enable_dns": true,
  "endpoint_service_name": "test123",
  "endpoint_service_id": "test123",
  "project_id": "6e9dfd51d1124e8d8498dce894923a0d",
  "whitelist": [ "192.168.1.1", "192.168.1.2" ],
  "enable_whitelist": true,
  "created_at": "2022-04-14T09:35:47Z",
  "updated_at": "2022-04-14T09:36:47Z",
  "tags": [ {
    "key": "test1",
    "value": "test1"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新终端节点，设置白名单信息为192.168.1.1、192.168.1.2，开启网络ACL隔离。

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

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

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

public class UpdateEndpointWhiteSolution {

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

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

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

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

UpdateEndpointWhiteRequest request = new UpdateEndpointWhiteRequest();
request.withVpcEndpointId("{vpc_endpoint_id}");
UpdateEndpointWhiteRequestBody body = new UpdateEndpointWhiteRequestBody();
List<String> listbodyWhitelist = new ArrayList<>();
listbodyWhitelist.add("192.168.1.1");
listbodyWhitelist.add("192.168.1.2");
body.withEnableWhitelist(true);
body.withWhitelist(listbodyWhitelist);
request.withBody(body);
try {
    UpdateEndpointWhiteResponse response = client.updateEndpointWhite(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

更新终端节点，设置白名单信息为192.168.1.1、192.168.1.2，开启网络ACL隔离。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudskvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskvpcep.v1 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateEndpointWhiteRequest()
        request.vpc_endpoint_id = "{vpc_endpoint_id}"
```

```
listWhitelistbody = [
    "192.168.1.1",
    "192.168.1.2"
]
request.body = UpdateEndpointWhiteRequestBody(
    enable_whitelist=True,
    whitelist=listWhitelistbody
)
response = client.update_endpoint_white(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

更新终端节点，设置白名单信息为192.168.1.1、192.168.1.2，开启网络ACL隔离。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEndpointWhiteRequest{}
    request.VpcEndpointId = "{vpc_endpoint_id}"
    var listWhitelistbody = []string{
        "192.168.1.1",
        "192.168.1.2",
    }
    enableWhitelistUpdateEndpointWhiteRequestBody := true
    request.Body = &model.UpdateEndpointWhiteRequestBody{
        EnableWhitelist: &enableWhitelistUpdateEndpointWhiteRequestBody,
        Whitelist: &listWhitelistbody,
    }
    response, err := client.UpdateEndpointWhite(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

```
}  
}
```

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.6 修改终端节点的路由表

功能介绍

修改终端节点的路由表。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoints/{vpc_endpoint_id}/routetables

表 4-130 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vpc_endpoint_id	是	String	终端节点的ID。

请求参数

表 4-131 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-132 请求 Body 参数

参数	是否必选	参数类型	描述
routetables	是	Array of strings	路由表ID列表。

响应参数

状态码： 200

表 4-133 响应 Body 参数

参数	参数类型	描述
routetables	Array of strings	路由表ID列表。 若未指定，返回默认VPC下路由表ID。 更新Gateway类型终端节点服务的终端节点时，显示此参数。
error	Array of RoutetableInfoError objects	当修改终端节点子网路由表失败时，返回错误提示信息

表 4-134 RoutetableInfoError

参数	参数类型	描述
bind_failed	Array of RoutetableInfoErrorDetail objects	绑定终端节点子网路由表失败信息。

参数	参数类型	描述
unbind_failed	Array of RoutetableInfoErrorDetail objects	解绑终端节点子网路由表失败信息。

表 4-135 RoutetableInfoErrorDetail

参数	参数类型	描述
id	String	路由表ID。
error_message	String	详细错误信息。

请求示例

修改终端节点的路由表

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoints/4189d3c2-8882-4871-a3c2-d380272eed83/routetables
{
  "routetables": [ "705290f3-0d00-41f2-aedc-71f09844e879" ]
}
```

响应示例

状态码： 200

服务器已成功处理了请求

```
{
  "routetables": [ "705290f3-0d00-41f2-aedc-71f09844e879" ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改终端节点的路由表

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

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

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

public class UpdateEndpointRoutetableSolution {
```

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

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

    VpcepClient client = VpcepClient.newBuilder()
        .withCredential(auth)
        .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
        .build();
    UpdateEndpointRoutetableRequest request = new UpdateEndpointRoutetableRequest();
    request.withVpcEndpointId("{vpc_endpoint_id}");
    UpdateEndpointRoutetableRequestBody body = new UpdateEndpointRoutetableRequestBody();
    List<String> listbodyRoutetables = new ArrayList<>();
    listbodyRoutetables.add("705290f3-0d00-41f2-aedc-71f09844e879");
    body.withRoutetables(listbodyRoutetables);
    request.withBody(body);
    try {
        UpdateEndpointRoutetableResponse response = client.updateEndpointRoutetable(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

修改终端节点的路由表

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpc.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpc.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = VpcepClient.new_builder() \
```

```
.with_credentials(credentials) \  
.with_region(VpcepRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = UpdateEndpointRoutetableRequest()  
    request.vpc_endpoint_id = "{vpc_endpoint_id}"  
    listRoutetablesbody = [  
        "705290f3-0d00-41f2-aedc-71f09844e879"  
    ]  
    request.body = UpdateEndpointRoutetableRequestBody(  
        routetables=listRoutetablesbody  
    )  
    response = client.update_endpoint_routetable(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

修改终端节点的路由表

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := vpcep.NewVpcepClient(  
        vpcep.VpcepClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateEndpointRoutetableRequest{}  
    request.VpcEndpointId = "{vpc_endpoint_id}"  
    var listRoutetablesbody = []string{  
        "705290f3-0d00-41f2-aedc-71f09844e879",  
    }  
    request.Body = &model.UpdateEndpointRoutetableRequestBody{  
        Routetables: listRoutetablesbody,  
    }  
    response, err := client.UpdateEndpointRoutetable(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {
```

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

更多

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

状态码

状态码	描述
200	服务器已成功处理了请求

错误码

请参见[错误码](#)。

4.3.7 修改终端节点策略

功能介绍

修改终端节点策略。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/{project_id}/vpc-endpoints/{vpc_endpoint_id}/policy

表 4-136 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
vpc_endpoint_id	是	String	终端节点的ID。

请求参数

表 4-137 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)。
Content-Type	否	String	发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。

表 4-138 请求 Body 参数

参数	是否必选	参数类型	描述
policy_statement	否	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。
policy_document	否	Object	终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）

表 4-139 PolicyStatement

参数	是否必选	参数类型	描述
Effect	是	String	<ul style="list-style-type: none">Allow，允许控制访问权限Deny，拒绝控制访问权限
Action	是	Array of strings	obs访问权限
Resource	是	Array of strings	obs对象

响应参数

状态码： 200

表 4-140 响应 Body 参数

参数	参数类型	描述
id	String	终端节点的ID，唯一标识。
service_type	String	终端节点连接的终端节点服务类型。 <ul style="list-style-type: none">• gateway: 由运维人员配置。用户无需创建，可直接使用。• interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中，运维人员配置的云服务无需创建，用户可直接使用。您可以通过查询公共终端节点服务列表，查看由运维人员配置的所有用户可见且可连接的终端节点服务，并通过创建终端节点服务创建 Interface类型的终端节点服务。
status	String	终端节点的连接状态。 <ul style="list-style-type: none">• pendingAcceptance: 待接受• creating: 创建中• accepted: 已接受• rejected: 已拒绝• failed: 失败• deleting: 删除中
active_status	Array of strings	账号状态。 <ul style="list-style-type: none">• frozen: 冻结• active: 解冻
endpoint_service_name	String	终端节点服务的名称。
marker_id	Integer	终端节点的报文标识。
endpoint_service_id	String	终端节点服务的ID。

参数	参数类型	描述
ip	String	访问所连接的终端节点服务的IP。 仅当同时满足如下条件时，返回该参数： 当查询连接interface类型终端节点服务的终端节点时。 终端节点服务启用“连接审批”功能，且已经“接受”连接审批。 “status”可以是“accepted”或者“rejected（仅支持“接受”连接审批后再“拒绝”的情况）”。
vpc_id	String	终端节点所在的VPC的ID。
created_at	String	终端节点的创建时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
updated_at	String	终端节点的更新时间。 采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ
project_id	String	项目ID，获取方法请参见获取项目ID。
tags	Array of TagList objects	标签列表，没有标签默认为空数组。
error	Array of QueryError objects	错误信息。 当终端节点状态异常，即“status”的值为“failed”时，会返回该字段。
whitelist	Array of strings	控制访问终端节点的白名单。 若未创建，则返回空列表。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
enable_whitelist	Boolean	是否开启网络ACL隔离。 <ul style="list-style-type: none">• true：开启网络ACL隔离• false：不开启网络ACL隔离 若未指定，则返回false。 创建连接Interface类型终端节点服务的终端节点时，显示此参数。
routetables	Array of strings	路由表ID列表。 若未指定，返回默认VPC下路由表ID。 创建gateway类型终端节点服务的终端节点时，显示此参数。
description	String	描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。

参数	参数类型	描述
policy_statement	Array of PolicyStatement objects	Gateway类型终端节点策略信息，仅限 OBS、SFS的终端节点服务的 enable_policy值为true时支持该参数。
policy_document	Object	终端节点策略信息，仅当终端节点服务的 enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）
endpoint_pool_id	String	待废弃，实例相关联的集群ID
public_border_group	String	终端节点关联的Public Border Group信息，只有当终端节点和边缘Pool相关联时才会返回该字段

表 4-141 TagList

参数	参数类型	描述
key	String	键。 key不能为空，长度1~128个字符（中文也可以输入128个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。 _sys_开头属于系统标签，租户不能输入。 key两头不能有空格字符。
value	String	值。 长度0~255个字符（中文也可以输入255个字符）。 可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。 资源标签值可以为空字符串。

表 4-142 QueryError

参数	参数类型	描述
error_code	String	错误编码。
error_message	String	错误信息。

表 4-143 PolicyStatement

参数	参数类型	描述
Effect	String	<ul style="list-style-type: none"> Allow, 允许控制访问权限 Deny, 拒绝控制访问权限
Action	Array of strings	obs访问权限
Resource	Array of strings	obs对象

请求示例

- 修改网关型终端节点策略，设置OBS访问权限和OBS对象并允许访问。

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoints/938c8167-631e-40a4-99f9-493753fbd16b/policy
{
  "policy_statement": [ {
    "Action": [ "obs:*:*" ],
    "Resource": [ "obs:*:*/*", "obs:*:*:*" ],
    "Effect": "Allow"
  } ]
}
```

- 修改终端节点策略信息。

```
PUT https://{endpoint}/v1/{project_id}/vpc-endpoints/2af6c328-057a-4146-882b-6828e270e594/policy
{
  "policy_document": {
    "Version": "5.0",
    "Statement": [ {
      "Effect": "Allow",
      "Principal": "*",
      "Action": [ "*" ],
      "Resource": [ "*" ]
    } ]
  }
}
```

响应示例

状态码： 200

服务器已成功处理了请求

- ```
{
 "id": "938c8167-631e-40a4-99f9-493753fbd16b",
 "status": "accepted",
 "tags": [],
 "marker_id": "302035929",
 "active_status": ["active"],
 "vpc_id": "0da03835-1dcf-4361-9b87-34139d58dd59",
 "service_type": "gateway",
 "project_id": "0605767a3300d5762fb7c0186d9e1779",
 "routetables": ["99477d3b-87f6-49d2-8f3b-2ffc72731a38"],
 "created_at": "2022-08-03T03:03:54Z",
 "updated_at": "2022-08-03T03:03:57Z",
 "endpoint_service_id": "4651bc78-5cec-41b7-b448-f77326ebbed0",
 "endpoint_service_name": "br-abc-aaa1.obs_test.4651bc78-5cec-41b7-b448-f77326ebbed0",
 "policy_statement": [{
 "Action": ["obs:*:*"],
 "Resource": ["obs:*:*/*", "obs:*:*:*"],
 }]
}
```

```
 "Effect" : "Allow"
 }],
 "description" : "",
 "endpoint_pool_id" : "b0ad6a4f-55c0-43f1-a26d-278639661fc2"
}

• {
 "id" : "2af6c328-057a-4146-882b-6828e270e594",
 "status" : "accepted",
 "tags" : [],
 "marker_id" : 335620220,
 "active_status" : ["active"],
 "vpc_id" : "a8494c3b-5dc9-46a4-8ca1-17a54f588d17",
 "service_type" : "interface",
 "project_id" : "0df9cbd6cf00f37***0c64b5af21",
 "created_at" : "2022-08-03T03:03:54Z",
 "updated_at" : "2022-08-03T03:03:57Z",
 "endpoint_service_id" : "925b47bd-ffef-4abe-a450-d997f4f4e2be",
 "endpoint_service_name" : "com.myhuaweicloud.cn-southwest-2.swr",
 "policy_document" : {
 "Version" : "5.0",
 "Statement" : [{
 "Effect" : "Allow",
 "Principal" : "*",
 "Action" : ["*"],
 "Resource" : ["*"]
 }]
 },
 "description" : "",
 "endpoint_pool_id" : "15027791-b2b5-4b12-8e01-8dec1b576bf1"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 修改网关型终端节点策略，设置OBS访问权限和OBS对象并允许访问。

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

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

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

public class UpdateEndpointPolicySolution {

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

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

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

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
UpdateEndpointPolicyRequest request = new UpdateEndpointPolicyRequest();
request.withVpcEndpointId("{vpc_endpoint_id}");
UpdateEndpointPolicyRequestBody body = new UpdateEndpointPolicyRequestBody();
List<String> listPolicyStatementResource = new ArrayList<>();
listPolicyStatementResource.add("obs:*:*/*/*");
listPolicyStatementResource.add("obs:*:*:*");
List<String> listPolicyStatementAction = new ArrayList<>();
listPolicyStatementAction.add("obs:*:*");
List<PolicyStatement> listbodyPolicyStatement = new ArrayList<>();
listbodyPolicyStatement.add(
 new PolicyStatement()
 .withEffect(PolicyStatement.EffectEnum.fromValue("Allow"))
 .withAction(listPolicyStatementAction)
 .withResource(listPolicyStatementResource)
);
body.withPolicyStatement(listbodyPolicyStatement);
request.withBody(body);
try {
 UpdateEndpointPolicyResponse response = client.updateEndpointPolicy(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

- 修改终端节点策略信息。

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

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

public class UpdateEndpointPolicySolution {

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

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

```
 .withSk(sk);

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
UpdateEndpointPolicyRequest request = new UpdateEndpointPolicyRequest();
request.withVpcEndpointId("{vpc_endpoint_id}");
UpdateEndpointPolicyRequestBody body = new UpdateEndpointPolicyRequestBody();
body.withPolicyDocument("{\"Version\":\"5.0\",\"Statement\":[{\"Action\":\"*\",\"Resource\":[\"*\"],\"Effect\":\"Allow\",\"Principal\":\"*\"}]\"}");
request.withBody(body);
try {
 UpdateEndpointPolicyResponse response = client.updateEndpointPolicy(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
}
```

## Python

- 修改网关型终端节点策略，设置OBS访问权限和OBS对象并允许访问。

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = UpdateEndpointPolicyRequest()
 request.vpc_endpoint_id = "{vpc_endpoint_id}"
 listResourcePolicyStatement = [
 "obs:*:*/*/*",
 "obs:*:*:*"
]
 listActionPolicyStatement = [
 "obs:*:*"
]
 listPolicyStatementbody = [
```

```
 PolicyStatement(
 effect="Allow",
 action=listActionPolicyStatement,
 resource=listResourcePolicyStatement
)
]
 request.body = UpdateEndpointPolicyRequestBody(
 policy_statement=listPolicyStatementbody
)
 response = client.update_endpoint_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)
```

- 修改终端节点策略信息。

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = UpdateEndpointPolicyRequest()
 request.vpc_endpoint_id = "{vpc_endpoint_id}"
 request.body = UpdateEndpointPolicyRequestBody(
 policy_document="{\"Version\": \"5.0\", \"Statement\": [{\"Action\": [\"*\"], \"Resource\": [\"*\"], \"Effect\": \"Allow\", \"Principal\": \"*\"}]}"
)
 response = client.update_endpoint_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

- 修改网关型终端节点策略，设置OBS访问权限和OBS对象并允许访问。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateEndpointPolicyRequest{}
 request.VpcEndpointId = "{vpc_endpoint_id}"
 var listResourcePolicyStatement = []string{
 "obs:*:*:*/*",
 "obs:*:*:*",
 }
 var listActionPolicyStatement = []string{
 "obs:*:*",
 }
 var listPolicyStatementbody = []model.PolicyStatement{
 {
 Effect: model.GetPolicyStatementEffectEnum().ALLOW,
 Action: listActionPolicyStatement,
 Resource: listResourcePolicyStatement,
 },
 }
 request.Body = &model.UpdateEndpointPolicyRequestBody{
 PolicyStatement: &listPolicyStatementbody,
 }
 response, err := client.UpdateEndpointPolicy(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 修改终端节点策略信息。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

```
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

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

client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.UpdateEndpointPolicyRequest{
 request.VpcEndpointId = "{vpc_endpoint_id}"
 var policyDocumentUpdateEndpointPolicyRequestBody interface{} = "{\"Version
 \": \"5.0\", \"Statement\": [{ \"Action\": [\"*\"], \"Resource\": [\"*\"], \"Effect\": \"Allow\", \"Principal\": [\"*\"]}]}\"
 request.Body = &model.UpdateEndpointPolicyRequestBody{
 PolicyDocument: &policyDocumentUpdateEndpointPolicyRequestBody,
 }
}
response, err := client.UpdateEndpointPolicy(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

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

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 服务器已成功处理了请求 |

## 错误码

请参见[错误码](#)。

### 4.3.8 删除网关型终端节点策略（待下线）

#### 功能介绍

删除网关型终端节点策略，该接口待下线，不建议使用。

#### 调用方法

请参见[如何调用API](#)。



## URI

DELETE /v1/{project\_id}/vpc-endpoints/{vpc\_endpoint\_id}/policy

表 4-144 路径参数

| 参数              | 是否必选 | 参数类型   | 描述       |
|-----------------|------|--------|----------|
| project_id      | 是    | String | 项目ID     |
| vpc_endpoint_id | 是    | String | 终端节点的ID。 |

## 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)。                  |
| Content-Type | 否    | String | 发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。 |

## 响应参数

状态码： 200

表 4-146 响应 Body 参数

| 参数 | 参数类型   | 描述            |
|----|--------|---------------|
| id | String | 终端节点的ID，唯一标识。 |

| 参数                    | 参数类型             | 描述                                                                                                                                                                                                                                                          |
|-----------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| service_type          | String           | 终端节点连接的终端节点服务类型。 <ul style="list-style-type: none"><li>gateway: 由运维人员配置。用户无需创建, 可直接使用。</li><li>interface: 包括运维人员配置的云服务和用户自己创建的私有服务。其中, 运维人员配置的云服务无需创建, 用户可直接使用。您可以通过查询公共终端节点服务列表, 查看由运维人员配置的所有用户可见且可连接的终端节点服务, 并通过创建终端节点服务创建 Interface 类型的终端节点服务。</li></ul> |
| status                | String           | 终端节点的连接状态。 <ul style="list-style-type: none"><li>pendingAcceptance: 待接受</li><li>creating: 创建中</li><li>accepted: 已接受</li><li>rejected: 已拒绝</li><li>failed: 失败</li><li>deleting: 删除中</li></ul>                                                                |
| active_status         | Array of strings | 账号状态。 <ul style="list-style-type: none"><li>frozen: 冻结</li><li>active: 解冻</li></ul>                                                                                                                                                                         |
| endpoint_service_name | String           | 终端节点服务的名称。                                                                                                                                                                                                                                                  |
| marker_id             | Integer          | 终端节点的报文标识。                                                                                                                                                                                                                                                  |
| endpoint_service_id   | String           | 终端节点服务的ID。                                                                                                                                                                                                                                                  |
| ip                    | String           | 访问所连接的终端节点服务的IP。<br>仅当同时满足如下条件时, 返回该参数:<br>当查询连接interface类型终端节点服务的终端节点时。<br>终端节点服务启用“连接审批”功能, 且已经“接受”连接审批。<br>“status”可以是“accepted”或者“rejected (仅支持“接受”连接审批后再“拒绝”的情况)”。                                                                                     |
| vpc_id                | String           | 终端节点所在的VPC的ID。                                                                                                                                                                                                                                              |

| 参数               | 参数类型                                             | 描述                                                                                                                                                            |
|------------------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| created_at       | String                                           | 终端节点的创建时间。<br>采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ                                                                                                              |
| updated_at       | String                                           | 终端节点的更新时间。<br>采用UTC时间格式，格式为：YYYY-MM-DDTHH:MM:SSZ                                                                                                              |
| project_id       | String                                           | 项目ID，获取方法请参见获取项目ID。                                                                                                                                           |
| tags             | Array of <a href="#">TagList</a> objects         | 标签列表，没有标签默认为空数组。                                                                                                                                              |
| error            | Array of <a href="#">QueryError</a> objects      | 错误信息。<br>当终端节点状态异常，即“status”的值为“failed”时，会返回该字段。                                                                                                              |
| whitelist        | Array of strings                                 | 控制访问终端节点的白名单。<br>若未创建，则返回空列表。<br>创建连接Interface类型终端节点服务的终端节点时，显示此参数。                                                                                           |
| enable_whitelist | Boolean                                          | 是否开启网络ACL隔离。 <ul style="list-style-type: none"><li>• true：开启网络ACL隔离</li><li>• false：不开启网络ACL隔离</li></ul> 若未指定，则返回false。<br>创建连接Interface类型终端节点服务的终端节点时，显示此参数。 |
| routetables      | Array of strings                                 | 路由表ID列表。<br>若未指定，返回默认VPC下路由表ID。<br>创建gateway类型终端节点服务的终端节点时，显示此参数。                                                                                             |
| description      | String                                           | 描述字段，支持中英文字母、数字等字符，不支持“<”或“>”字符。                                                                                                                              |
| policy_statement | Array of <a href="#">PolicyStatement</a> objects | Gateway类型终端节点策略信息，仅限OBS、SFS的终端节点服务的enable_policy值为true时支持该参数。                                                                                                 |
| policy_document  | Object                                           | 终端节点策略信息，仅当终端节点服务的enable_policy值为true时支持该参数，默认值为完全访问权限。（OBS、SFS的终端节点服务暂不支持该参数）                                                                                |
| endpoint_pool_id | String                                           | 待废弃，实例相关联的集群ID                                                                                                                                                |

| 参数                  | 参数类型   | 描述                                                     |
|---------------------|--------|--------------------------------------------------------|
| public_border_group | String | 终端节点关联的Public Border Group信息，只有当终端节点和边缘Pool相关联时才会返回该字段 |

表 4-147 TagList

| 参数    | 参数类型   | 描述                                                                                                                                                |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 键。<br>key不能为空，长度1~128个字符（中文也可以输入128个字符）。<br>可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : = + - @。<br>_sys_开头属于系统标签，租户不能输入。<br>key两头不能有空格字符。 |
| value | String | 值。<br>长度0~255个字符（中文也可以输入255个字符）。<br>可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格，以及以下字符：_ . : / = + - @。<br>资源标签值可以为空字符串。                                |

表 4-148 QueryError

| 参数            | 参数类型   | 描述    |
|---------------|--------|-------|
| error_code    | String | 错误编码。 |
| error_message | String | 错误信息。 |

表 4-149 PolicyStatement

| 参数       | 参数类型             | 描述                                                                                     |
|----------|------------------|----------------------------------------------------------------------------------------|
| Effect   | String           | <ul style="list-style-type: none"><li>Allow, 允许控制访问权限</li><li>Deny, 拒绝控制访问权限</li></ul> |
| Action   | Array of strings | obs访问权限                                                                                |
| Resource | Array of strings | obs对象                                                                                  |

## 请求示例

删除网关型终端节点策略

```
DELETE https://{endpoint}/v1/{project_id}/vpc-endpoints/938c8167-631e-40a4-99f9-493753fbd16b/policy
```

## 响应示例

状态码： 200

服务器已成功处理了请求

```
{
 "id": "938c8167-631e-40a4-99f9-493753fbd16b",
 "status": "accepted",
 "tags": [],
 "jobId": "endpoint_add_938c8167-631e-40a4-99f9-493753fbd16b",
 "marker_id": "302035929",
 "active_status": ["active"],
 "vpc_id": "0da03835-1dcf-4361-9b87-34139d58dd59",
 "service_type": "gateway",
 "project_id": "0605767a3300d5762fb7c0186d9e1779",
 "routetables": ["99477d3b-87f6-49d2-8f3b-2ffc72731a38"],
 "created_at": "2022-08-03T03:03:54Z",
 "updated_at": "2022-08-03T03:03:57Z",
 "endpoint_service_id": "4651bc78-5cec-41b7-b448-f77326ebbed0",
 "endpoint_service_name": "br-abc-aaa1.obs_test.4651bc78-5cec-41b7-b448-f77326ebbed0",
 "specification_name": "default",
 "enable_status": "enable",
 "description": "",
 "endpoint_pool_id": "b0ad6a4f-55c0-43f1-a26d-278639661fc2"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

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

public class DeleteEndpointPolicySolution {

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

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

```
 .withSk(sk);

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
DeleteEndpointPolicyRequest request = new DeleteEndpointPolicyRequest();
request.withVpcEndpointId("{vpc_endpoint_id}");
try {
 DeleteEndpointPolicyResponse response = client.deleteEndpointPolicy(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = DeleteEndpointPolicyRequest()
 request.vpc_endpoint_id = "{vpc_endpoint_id}"
 response = client.delete_endpoint_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"
vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

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

## 更多

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

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 服务器已成功处理了请求 |

## 错误码

请参见[错误码](#)。

## 4.4 资源配额功能

## 4.4.1 查询配额

### 功能介绍

查询用户的资源配额，包括终端节点服务和终端节点。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/quotas

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。 |

表 4-151 Query 参数

| 参数   | 是否必选 | 参数类型   | 描述                                                                                                    |
|------|------|--------|-------------------------------------------------------------------------------------------------------|
| type | 否    | String | 资源类型。 <ul style="list-style-type: none"><li>endpoint_service: 终端节点服务</li><li>endpoint: 终端节点</li></ul> |

### 请求参数

表 4-152 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)。                  |
| Content-Type | 否    | String | 发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。 |



## 响应参数

状态码： 200

表 4-153 响应 Body 参数

| 参数     | 参数类型                         | 描述          |
|--------|------------------------------|-------------|
| quotas | ResourcesResponseBody object | 查询配额接口响应结构体 |

表 4-154 ResourcesResponseBody

| 参数        | 参数类型                    | 描述   |
|-----------|-------------------------|------|
| resources | Array of Quotas objects | 资源列表 |

表 4-155 Quotas

| 参数    | 参数类型    | 描述                                                                                                                        |
|-------|---------|---------------------------------------------------------------------------------------------------------------------------|
| type  | String  | 资源类型。支持根据资源类型过滤查询指定类型的配额。 <ul style="list-style-type: none"><li>endpoint_service: 终端节点服务</li><li>endpoint: 终端节点</li></ul> |
| used  | Integer | 已创建的资源个数。<br>取值范围：0~quota数。                                                                                               |
| quota | Integer | 资源的最大配额数。<br>取值范围：各类型资源默认配额数的最大值。                                                                                         |

## 请求示例

- 查询终端节点服务配额  
GET https://{endpoint}/v1/{project\_id}/quotas?type=endpoint\_service
- 查询终端节点配额  
GET https://{endpoint}/v1/{project\_id}/quotas?type=endpoint

## 响应示例

状态码： 200

服务器已成功处理了请求

```
{
 "quotas": {
```

```
"resources": [{
 "type": "endpoint",
 "used": 4,
 "quota": 150
}, {
 "type": "endpoint_service",
 "used": 10,
 "quota": 100
}
]
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

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

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

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

 VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
 ListQuotaDetailsRequest request = new ListQuotaDetailsRequest();
 try {
 ListQuotaDetailsResponse response = client.listQuotaDetails(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

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

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
```

```
Build()

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

## 更多

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

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 服务器已成功处理了请求 |

## 错误码

请参见[错误码](#)。

# 4.5 TAG 功能

## 4.5.1 查询资源实例接口

### 功能介绍

使用标签过滤查询租户下资源的实例。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/{resource\_type}/resource\_instances/action

表 4-156 路径参数

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

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                          |
|---------------|------|--------|-------------------------------------------------------------------------------------------------------------|
| resource_type | 是    | String | 资源类型，值为：<br>endpoint_service或endpoint。<br>- endpoint_service：云服务器器，适用于作为服务器使用。 - endpoint：虚拟IP，适用于作为虚IP场景使用 |

## 请求参数

表 4-157 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)。                  |
| Content-Type | 否    | String | 发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。 |

表 4-158 请求 Body 参数

| 参数   | 是否必选 | 参数类型                                           | 描述                                                                                                                                                                          |
|------|------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tags | 否    | Array of <a href="#">TagValuesList</a> objects | 包含标签，最多包含10个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，<br>同一个key中values不能重复。<br>结果返回包含所有标签的资源列表，<br>key之间是与的关系，key-value结构中value是或的关系。<br>无tag过滤条件时返回全量数据。 |

| 参数           | 是否必选 | 参数类型                                           | 描述                                                                                                                                                                             |
|--------------|------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tags_any     | 否    | Array of <a href="#">TagValuesList</a> objects | 包含任意标签，最多包含10个key，每个key下面的value最多10个，<br>每个key对应的value可以为空数组但结构体不能缺失。<br>Key不能重复，同一个key中values不能重复。<br>结果返回包含标签的资源列表，key之间是或的关系，key、value结构中value是或的关系。<br>无过滤条件时返回全量数据。       |
| not_tags     | 否    | Array of <a href="#">TagValuesList</a> objects | 不包含标签，最多包含10个key，每个key下面的value最多10个，<br>每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，<br>同一个key中values不能重复。<br>结果返回不包含标签的资源列表，key之间是与的关系，<br>key、value结构中value是或的关系。<br>无过滤条件时返回全量数据。   |
| not_tags_any | 否    | Array of <a href="#">TagValuesList</a> objects | 不包含任意标签，最多包含10个key，每个key下面的value最多10个，<br>每个key对应的value可以为空数组但结构体不能缺失。Key不能重复，<br>同一个key中values不能重复。<br>结果返回不包含标签的资源列表，<br>key之间是与的关系，key、value结构中value是或的关系。<br>无过滤条件时返回全量数据。 |
| sys_tags     | 否    | Array of <a href="#">TagValuesList</a> objects | 系统标签，                                                                                                                                                                          |
| limit        | 否    | String                                         | 查询记录数（action为count时无此参数）如果action为filter默认为1000，<br>limit最多为1000，不能为负数，最小值为1。                                                                                                   |

| 参数              | 是否必选 | 参数类型                   | 描述                                                                                                                                                                                                                                            |
|-----------------|------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| offset          | 否    | String                 | 索引位置，偏移量（action为count时无此参数）从第一条数据偏移offset条数据后开始查询，<br>如果action为filter默认为0（偏移0条数据，表示从第一条数据开始查询），<br>必须为数字，不能为负数。                                                                                                                               |
| action          | 是    | String                 | 操作标识（仅限于filter，count）：filter（过滤），<br>count(查询总条数)如果是filter就按照过滤条件查询，<br>如果是count，只需要返回总条数，禁止返回其他字段。                                                                                                                                           |
| matches         | 否    | Array of Match objects | 搜索字段，key为要匹配的字段，如resource_name等。<br>value为匹配的值。<br>key为固定字典值，不能包含重复的key或不支持的key。<br>根据key的值确认是否需要模糊匹配，如resource_name默认为模糊搜索（不区分大小写），<br>如果value为空字符串精确匹配（多数服务不存在资源名称为空的情况，<br>因此此类情况返回空列表）。<br>resource_id为精确匹配。<br>第一期只做resource_name，后续再扩展。 |
| without_any_tag | 否    | Boolean                | 默认为false，取值【true/false】，当withoutAnyTag=true，忽略tags、tagsAny、notTags、notTagsAny参数校验。                                                                                                                                                            |

表 4-159 TagValuesList

| 参数     | 是否必选 | 参数类型             | 描述                                                                                                                                                     |
|--------|------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| key    | 是    | String           | 键。<br>最大长度128个unicode字符。<br>key不能为空。(搜索时不对此参数做字符集校),<br>key不能为空或者空字符串,不能为空格,校验和使用之前先trim前后半角空格。                                                        |
| values | 是    | Array of strings | 值列表。<br>每个值最大长度255个unicode字符,校验和使用之前先trim前后半角空格。<br>value可为空数组但不可缺省。<br>如果values为空列表,则表示any_value(查询任意value)。<br>value之间为或的关系。(搜索时不对此参数做字符集校验,只做长度校验)。 |

表 4-160 Match

| 参数    | 是否必选 | 参数类型   | 描述                              |
|-------|------|--------|---------------------------------|
| key   | 是    | String | 键。第一期限定为resource_name, 后续扩展。    |
| value | 是    | String | 值。每个值最大长度255个unicode字符。不校验字符集范。 |

## 响应参数

状态码： 200

表 4-161 响应 Body 参数

| 参数          | 参数类型                                              | 描述     |
|-------------|---------------------------------------------------|--------|
| resources   | Array of <a href="#">ResourceInstance</a> objects | 资源实例详情 |
| total_count | Integer                                           | 总记录数。  |



表 4-162 ResourceInstance

| 参数            | 参数类型                                     | 描述                                     |
|---------------|------------------------------------------|----------------------------------------|
| resource_id   | String                                   | 资源ID, Endpoint Service ID或Endpoint ID。 |
| tags          | Array of <a href="#">TagList</a> objects | 标签列表。                                  |
| resource_name | String                                   | 资源名称, 资源没有名称时, 返回ID。                   |

表 4-163 TagList

| 参数    | 参数类型   | 描述                                                                                                                                                    |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 键。<br>key不能为空, 长度1~128个字符(中文也可以输入128个字符)。<br>可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格, 以及以下字符: _ . : = + - @。<br>_sys_开头属于系统标签, 租户不能输入。<br>key两头不能有空格字符。 |
| value | String | 值。<br>长度0~255个字符(中文也可以输入255个字符)。<br>可用 UTF-8 格式表示的字母(包含中文、西班牙语、葡语等)、数字和空格, 以及以下字符: _ . : / = + - @。<br>资源标签值可以为空字符串。                                  |

## 请求示例

- 使用标签查询终端节点

POST https://{endpoint}/v1/{project\_id}/endpoint/resource\_instances/action

action为filter时请求体

```
{
 "offset": "100",
 "limit": "100",
 "action": "filter",
 "matches": [
 {
 "key": "resource_name",
 "value": "resource1"
 }
],
 "not_tags": [
 {
 "key": "key1",
```

```
 "values": [
 "*value1",
 "value2"
]
 },
],
"tags": [
 {
 "key": "key1",
 "values": [
 "*value1",
 "value2"
]
 }
],
"tags_any": [
 {
 "key": "key1",
 "values": [
 "value1",
 "value2"
]
 }
],
"not_tags_any": [
 {
 "key": "key1",
 "values": [
 "value1",
 "value2"
]
 }
]
]
```

- 查询终端节点服务资源总数

[https://{endpoint}/v1/{project\\_id}/endpoint\\_service/resource\\_instances/action](https://{endpoint}/v1/{project_id}/endpoint_service/resource_instances/action)

action为count时请求体

```
{
 "action": "count",
 "not_tags": [
 {
 "key": "key1",
 "values": [
 "value1",
 "*value2"
]
 }
],
 "tags": [
 {
 "key": "key1",
 "values": [
 "value1",
 "value2"
]
 },
 {
 "key": "key2",
 "values": [
 "value1",
 "value2"
]
 }
],
 "tags_any": [
 {
 "key": "key1",
 "values": [
```

```
 "value1",
 "value2"
]
 },
],
 "not_tags_any": [
 {
 "key": "key1",
 "values": [
 "value1",
 "value2"
]
 }
],
 "matches": [
 {
 "key": "resource_name",
 "value": "resource1"
 }
]
]
}
```

## 响应示例

**状态码： 200**

服务器已成功处理了请求

```
- action为filter时响应体
{
 "resources": [
 {
 "resource_id": "cdfs_cefs_wesas_12_dsad",
 "resource_name": "resource1",
 "tags": [
 {
 "key": "key1", "value": "value1"
 },
 {
 "key": "key2", "value": "value1"
 }
]
 },
],
 "total_count": 1000
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 使用标签查询终端节点

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

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

```
public class ListResourceInstancesSolution {

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

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

 VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
 ListResourceInstancesRequest request = new ListResourceInstancesRequest();

 request.withResourceType(ListResourceInstancesRequest.ResourceTypeEnum.fromValue("{resource_type}"));
 try {
 ListResourceInstancesResponse response = client.listResourceInstances(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

- 查询终端节点服务资源总数

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

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

public class ListResourceInstancesSolution {

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

 ICredential auth = new BasicCredentials()
```

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

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
ListResourceInstancesRequest request = new ListResourceInstancesRequest();

request.withResourceType(ListResourceInstancesRequest.ResourceTypeEnum.fromValue("{resource_type}"));
try {
 ListResourceInstancesResponse response = client.listResourceInstances(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

- 使用标签查询终端节点

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = ListResourceInstancesRequest()
 request.resource_type = "{resource_type}"
 response = client.list_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)
```

- 查询终端节点服务资源总数

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = ListResourceInstancesRequest()
 request.resource_type = "{resource_type}"
 response = client.list_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

- 使用标签查询终端节点

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

```
client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ListResourceInstancesRequest{}
request.ResourceType =
model.GetListResourceInstancesRequestResourceTypeEnum().RESOURCE_TYPE
response, err := client.ListResourceInstances(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

- 查询终端节点服务资源总数

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListResourceInstancesRequest{}
 request.ResourceType =
model.GetListResourceInstancesRequestResourceTypeEnum().RESOURCE_TYPE
response, err := client.ListResourceInstances(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

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

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 服务器已成功处理了请求 |

## 错误码

请参见[错误码](#)。

## 4.5.2 批量添加或删除资源标签接口

### 功能介绍

为指定Endpoint Service或Endpoint批量添加或删除标签。

- 一个资源上最多有10个标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v1/{project\_id}/{resource\_type}/{resource\_id}/tags/action

表 4-164 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                        |
|---------------|------|--------|-----------------------------------------------------------------------------------------------------------|
| project_id    | 是    | String | 项目ID。                                                                                                     |
| resource_type | 是    | String | 资源类型。 <ul style="list-style-type: none"><li>• endpoint_service: 终端节点服务</li><li>• endpoint: 终端节点</li></ul> |
| resource_id   | 是    | String | 资源ID, Endpoint Service ID或Endpoint ID。                                                                    |



## 请求参数

表 4-165 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)。                  |
| Content-Type | 否    | String | 发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。 |

表 4-166 请求 Body 参数

| 参数     | 是否必选 | 参数类型                                | 描述                                   |
|--------|------|-------------------------------------|--------------------------------------|
| tags   | 否    | Array of <b>ResourceTag</b> objects | 标签列表，没有标签默认为空数组。                     |
| action | 是    | String                              | 操作标识：仅限于<br>create（创建）<br>delete（删除） |

表 4-167 ResourceTag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                     |
|-------|------|--------|--------------------------------------------------------------------------------------------------------|
| key   | 是    | String | 键。最大长度128个unicode字符。<br>key需要满足标签字符集规范。                                                                |
| value | 否    | String | 值。action为create时必选，每个值最大长度255个unicode字符，删除时如果value有值按照key/value删除，如果value没值，则按照key删除。value需要满足标签字符集规范。 |

## 响应参数

无

## 请求示例

- 批量添加资源标签

POST https://{endpoint}/v1/{project\_id}/endpoint\_service/{resource\_id}/tags/action

```
{
 "action": "create",
 "tags": [{
 "key": "key1",
 "value": "value1"
 }, {
 "key": "key",
 "value": "value3"
 }]
}
```

- 批量删除资源标签

POST https://{endpoint}/v1/{project\_id}/endpoint/{resource\_id}/tags/action

```
{
 "action": "delete",
 "tags": [{
 "key": "key1"
 }, {
 "key": "key2",
 "value": "value3"
 }]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

- 批量添加资源标签

```
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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

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

public class BatchAddOrRemoveResourceInstanceSolution {

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

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

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

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
BatchAddOrRemoveResourceInstanceRequest request = new
BatchAddOrRemoveResourceInstanceRequest();

request.withResourceType(BatchAddOrRemoveResourceInstanceRequest.ResourceTypeEnum.fromValue(
"{resource_type}"));
request.withResourceId("{resource_id}");
BatchAddOrRemoveResourceInstanceRequestBody body = new
BatchAddOrRemoveResourceInstanceRequestBody();
List<ResourceTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
 new ResourceTag()
 .withKey("key1")
 .withValue("value1")
);
listbodyTags.add(
 new ResourceTag()
 .withKey("key")
 .withValue("value3")
);

body.withAction(BatchAddOrRemoveResourceInstanceRequestBody.ActionEnum.fromValue("create"));
body.withTags(listbodyTags);
request.withBody(body);
try {
 BatchAddOrRemoveResourceInstanceResponse response =
client.batchAddOrRemoveResourceInstance(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

- **批量删除资源标签**

```
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.vpcep.v1.region.VpcepRegion;
import com.huaweicloud.sdk.vpcep.v1.*;
import com.huaweicloud.sdk.vpcep.v1.model.*;

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

public class BatchAddOrRemoveResourceInstanceSolution {

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

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

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

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
BatchAddOrRemoveResourceInstanceRequest request = new
BatchAddOrRemoveResourceInstanceRequest();

request.withResourceType(BatchAddOrRemoveResourceInstanceRequest.ResourceTypeEnum.fromValue(
"{resource_type}"));
request.withResourceId("{resource_id}");
BatchAddOrRemoveResourceInstanceRequestBody body = new
BatchAddOrRemoveResourceInstanceRequestBody();
List<ResourceTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
 new ResourceTag()
 .withKey("key1")
);
listbodyTags.add(
 new ResourceTag()
 .withKey("key2")
 .withValue("value3")
);

body.withAction(BatchAddOrRemoveResourceInstanceRequestBody.ActionEnum.fromValue("delete"));
body.withTags(listbodyTags);
request.withBody(body);
try {
 BatchAddOrRemoveResourceInstanceResponse response =
client.batchAddOrRemoveResourceInstance(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

- 批量添加资源标签

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *
```

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

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = BatchAddOrRemoveResourceInstanceRequest()
 request.resource_type = "{resource_type}"
 request.resource_id = "{resource_id}"
 listTagsbody = [
 ResourceTag(
 key="key1",
 value="value1"
),
 ResourceTag(
 key="key",
 value="value3"
)
]
 request.body = BatchAddOrRemoveResourceInstanceRequestBody(
 action="create",
 tags=listTagsbody
)
 response = client.batch_add_or_remove_resource_instance(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 批量删除资源标签

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.v1 import *

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

 credentials = BasicCredentials(ak, sk, projectId)

 client = VpcepClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(VpcepRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
 request = BatchAddOrRemoveResourceInstanceRequest()
 request.resource_type = "{resource_type}"
 request.resource_id = "{resource_id}"
 listTagsbody = [
 ResourceTag(
 key="key1"
),
 ResourceTag(
 key="key2",
 value="value3"
)
]
 request.body = BatchAddOrRemoveResourceInstanceRequestBody(
 action="delete",
 tags=listTagsbody
)
 response = client.batch_add_or_remove_resource_instance(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

- 批量添加资源标签

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.BatchAddOrRemoveResourceInstanceRequest{}
 request.ResourceType =
 model.GetBatchAddOrRemoveResourceInstanceRequestResourceTypeEnum().RESOURCE_TYPE
 request.ResourceId = "{resource_id}"
 valueTags:= "value1"
 valueTags1:= "value3"
```

```
var listTagsbody = []model.ResourceTag{
 {
 Key: "key1",
 Value: &valueTags,
 },
 {
 Key: "key",
 Value: &valueTags1,
 },
}
request.Body = &model.BatchAddOrRemoveResourceInstanceRequestBody{
 Action: model.GetBatchAddOrRemoveResourceInstanceRequestBodyActionEnum().CREATE,
 Tags: &listTagsbody,
}
response, err := client.BatchAddOrRemoveResourceInstance(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

- 批量删除资源标签

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.BatchAddOrRemoveResourceInstanceRequest{}
 request.ResourceType =
model.GetBatchAddOrRemoveResourceInstanceRequestResourceTypeEnum().RESOURCE_TYPE
 request.ResourceId = "{resource_id}"
 valueTags := "value3"
 var listTagsbody = []model.ResourceTag{
 {
 Key: "key1",
 },
 {
 Key: "key2",
 Value: &valueTags,
 },
 }
}
```

```
request.Body = &model.BatchAddOrRemoveResourceInstanceRequestBody{
 Action: model.GetBatchAddOrRemoveResourceInstanceRequestBodyActionEnum().DELETE,
 Tags: &listTagsbody,
}
response, err := client.BatchAddOrRemoveResourceInstance(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

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

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 204 | 服务器已成功处理了请求 |

## 错误码

请参见[错误码](#)。

## 4.5.3 查询租户资源标签接口

### 功能介绍

根据租户ID和资源类型，获取租户下资源的标签。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v1/{project\_id}/{resource\_type}/tags

表 4-168 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                    |
|---------------|------|--------|-------------------------------------------------------------------------------------------------------|
| project_id    | 是    | String | 项目ID。                                                                                                 |
| resource_type | 是    | String | 资源类型。 <ul style="list-style-type: none"><li>endpoint_service: 终端节点服务</li><li>endpoint: 终端节点</li></ul> |



## 请求参数

表 4-169 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)。                  |
| Content-Type | 否    | String | 发送的实体的MIME类型。推荐用户默认使用application/json，如果API是对象、镜像上传等接口，媒体类型可按照流类型的不同进行确定。 |

## 响应参数

状态码： 200

表 4-170 响应 Body 参数

| 参数   | 参数类型                                           | 描述                                                                                                                                                                  |
|------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tags | Array of <a href="#">TagValuesList</a> objects | 包含标签，最多包含10个key，每个key下面的value最多10个，每个key对应的value可以为空数组但结构体不能缺失。<br>Key不能重复，同一个key中values不能重复。结果返回包含所有标签的资源列表，<br>key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 |

表 4-171 TagValuesList

| 参数  | 参数类型   | 描述                                                                                      |
|-----|--------|-----------------------------------------------------------------------------------------|
| key | String | 键。<br>最大长度128个unicode字符。key不能为空。(搜索时不对此参数做字符集校)，key不能为空或者空字符串，不能为空格，校验和使用之前先trim前后半角空格。 |

| 参数     | 参数类型             | 描述                                                                                                                                                 |
|--------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| values | Array of strings | 值列表。<br>每个值最大长度255个unicode字符，校验和使用之前先trim前后半角空格。<br>value可为空数组但不可缺省。如果values为空列表，则表示any_value（查询任意value）。<br>value之间为或的关系。（搜索时不对此参数做字符集校验，只做长度校验）。 |

## 请求示例

- 查询终端节点服务的资源标签  
GET https://{endpoint}/v1/{project\_id}/endpoint\_service/tags
- 查询终端节点的资源标签  
GET https://{endpoint}/v1/{project\_id}/endpoint/tags

## 响应示例

状态码： 200

标签列表。

```
{
 "tags" : [{
 "key" : "key1",
 "values" : ["value1", "value2"]
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

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

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

public class ListQueryProjectResourceTagsSolution {

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

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

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

VpcepClient client = VpcepClient.newBuilder()
 .withCredential(auth)
 .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
 .build();
ListQueryProjectResourceTagsRequest request = new ListQueryProjectResourceTagsRequest();

request.withResourceType(ListQueryProjectResourceTagsRequest.ResourceTypeEnum.fromValue("{resource_t
ype}"));
try {
 ListQueryProjectResourceTagsResponse response = client.listQueryProjectResourceTags(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkvpcep.v1.region.vpcep_region import VpcepRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkvpcep.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")
 projectId = "{project_id}"

 credentials = BasicCredentials(ak, sk, projectId)

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

 try:
 request = ListQueryProjectResourceTagsRequest()
 request.resource_type = "{resource_type}"
 response = client.list_query_project_resource_tags(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
```

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

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 vpcep "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/vpcep/v1/region"
)

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

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

 client := vpcep.NewVpcepClient(
 vpcep.VpcepClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListQueryProjectResourceTagsRequest{}
 request.ResourceType =
model.GetListQueryProjectResourceTagsRequestResourceTypeEnum().RESOURCE_TYPE
 response, err := client.ListQueryProjectResourceTags(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

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

## 状态码

| 状态码 | 描述    |
|-----|-------|
| 200 | 标签列表。 |

## 错误码

请参见[错误码](#)。

# 5 应用示例

## 5.1 示例：配置跨 VPC 通信的终端节点

### 场景描述

如果用户想要实现VPC间的资源通信，相比于通过公网IP的访问方式，VPC终端节点支持将VPC私密地连接到终端节点服务（云服务、用户私有服务），无需使用弹性公网IP，访问效率更高，组网更安全。

下面介绍如何通过调用[创建终端节点服务](#)API和[创建终端节点](#)API实现终端节点到终端节点服务的连接。API的调用方法请参见[如何调用API](#)。

#### 📖 说明

通过IAM服务获取到的Token有效期为24小时，需要使用同一个Token鉴权时，可以先将Token缓存，避免频繁调用。

### 前提条件

您需要规划VPC终端节点所在的区域信息，并根据区域确定调用API的Endpoint，详细信息请参见[终端节点（Endpoint）](#)。

### 创建终端节点服务

在本示例中，为云服务器创建了一个接口型的终端节点服务。

#### 📖 说明

创建终端节点服务前请先获取创建的必要信息，如服务后端资源的Port ID、所在的虚拟私有云（VPC）的VPC ID等，详情请参考[创建终端节点服务](#)。

```
{
 "port_id": "4189d3c2-8882-4871-a3c2-d380272eed88",
 "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed80",
 "approval_enabled": false,
 "service_type": "interface",
 "server_type": "VM",
 "ports":
 [
 {
```

```
 "client_port":8080,
 "server_port":80,
 "protocol":"TCP"
 },
 {
 "client_port":8081,
 "server_port":80,
 "protocol":"TCP"
 }
]
```

- port\_id: 标识终端节点服务后端资源的ID。例如创建云服务器类型的终端节点服务，则对应云服务器IP地址对应的网卡ID。
- vpc\_id: 终端节点服务对应后端资源所在的VPC的ID。
- approval\_enabled: 终端节点连接终端节点服务是否需要审批，例如“false”，表示可以直接连接，不需要审批。
- service\_type: 终端节点服务类型。例如“interface”，表示接口类型的终端节点服务。
- server\_type: 创建的终端节点服务的后端资源类型。例如“VM”，表示创建云服务器类型的终端节点服务。
- ports.client\_port: 终端节点提供给用户，作为访问终端节点服务的端口。
- ports.server\_port: 终端节点服务绑定了后端资源，作为提供服务的端口。
- ports.protocol: 端口映射协议。

## 创建终端节点

在本示例中，创建了一个终端节点，且支持自动创建内网域名。

### 📖 说明

创建终端节点前请先获取创建的必要信息，如终端节点所在的VPC的ID、Subnet ID和上一步创建返回的终端节点服务ID等，详情请参考[创建终端节点](#)。

```
{
 "subnet_id": "4189d3c2-8882-4871-a3c2-d380272eed81",
 "vpc_id": "4189d3c2-8882-4871-a3c2-d380272eed82",
 "endpoint_service_id": "4189d3c2-8882-4871-a3c2-d380272eed83",
 "enable_dns": true
}
```

- subnet\_id: 终端节点所在VPC的子网ID
- vpc\_id: 终端节点所在的VPC的ID。
- endpoint\_service\_id: 终端节点服务的ID。
- enable\_dns: 是否为终端节点创建内网域名。例如“true”，表示创建内网域名。

创建完成之后，终端节点所在的VPC内的实例可使用该终端节点访问ID为“4189d3c2-8882-4871-a3c2-d380272eed83”的终端节点服务。

# 6 附录

## 6.1 状态码

- 正常

表 6-1 正常时的返回值

| 状态码 | 返回值        | 状态码说明               |
|-----|------------|---------------------|
| 200 | OK         | POST、GET和PUT操作正常返回。 |
| 204 | No Content | DELETE操作正常返回。       |

- 异常

表 6-2 异常时的返回值

| 状态码 | 返回值                           | 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。               |

| 状态码 | 返回值                   | 状态码说明                     |
|-----|-----------------------|---------------------------|
| 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。               |
| 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。                     |

## 6.2 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 | EndPoint.0002 | Parameter error.                                           | 参数异常。         | 校验参数是否正确。         |
| 400 | EndPoint.0006 | Invalid limit.                                             | 无效的数量限制。      | 请输入正确的limit参数。    |
| 400 | EndPoint.0007 | Invalid action.                                            | 无效的操作。        | 请输入正确的action参数。   |
| 400 | EndPoint.0009 | The remote address does not match.                         | 访问远端地址不匹配。    | 请确认是否有权访问。        |
| 400 | EndPoint.0010 | Invalid offset.                                            | 无效的分页。        | 请输入正确的offset参数。   |
| 400 | EndPoint.0014 | Invalid project ID.                                        | project_id无效。 | 请输入正确的project_id。 |
| 400 | EndPoint.0015 | Invalid specification.                                     | 规格无效。         | 请输入正确的规格。         |
| 400 | EndPoint.0016 | The number of batch operated resources exceeded the limit. | 批量操作资源数量超过限制。 | 请缩小您批量操作资源的数量。    |



| 状态码 | 错误码           | 错误信息                                                         | 描述                    | 处理措施                   |
|-----|---------------|--------------------------------------------------------------|-----------------------|------------------------|
| 400 | EndPoint.0017 | Invalid sort_key.                                            | sort_key无效。           | 请输入正确的sort_key。        |
| 400 | EndPoint.0018 | Invalid sort_dir.                                            | sort_dir无效。           | 请输入正确的sort_dir。        |
| 400 | EndPoint.0019 | Invalid status.                                              | 状态无效。                 | 请输入正确的status。          |
| 400 | EndPoint.0020 | Invalid VPC ID.                                              | vpclId无效。             | 请输入正确的vpclId。          |
| 400 | EndPoint.0021 | Invalid marker_id.                                           | marker_id无效。          | 请输入正确的marker ID。       |
| 400 | EndPoint.0022 | The number of requests exceeded the limit. Please try later. | 请求次数太多，请稍后重试。         | 请稍后重试。                 |
| 400 | EndPoint.0023 | Invalid subnet_id.                                           | 无效的subnet_id。         | 请输入正确的subnet_id。       |
| 400 | EndPoint.1003 | Invalid service name.                                        | 服务名称无效。               | 请输入正确的服务名称。            |
| 400 | EndPoint.1004 | Invalid request.                                             | 创建终端节点请求无效。           | 请输入正确的请求体。             |
| 400 | EndPoint.1005 | DNS service Failed: Failed to create a DNS zone.             | DNS服务失败：创建DNS zone失败。 | 请联系技术支持。               |
| 400 | EndPoint.1008 | Failed to obtain the token.                                  | 获取token失败。            | 请联系技术支持。               |
| 400 | Endpoint.1018 | Quota exceeded.                                              | 申请资源数量超出配额。           | 请联系技术支持。               |
| 400 | EndPoint.1019 | Invalid route table ID.                                      | 无效的routetable ID。     | 请输入正确的routetable ID参数。 |
| 400 | EndPoint.2001 | The VPC does not exist.                                      | 当前用户下不存在该VPC。         | 请输入正确的当前用户下的VPC ID。    |
| 400 | EndPoint.2002 | The request input parameter is empty.                        | 请求入参为空。               | 参数不能不空，请输入正确的参数。       |

| 状态码 | 错误码           | 错误信息                                                           | 描述              | 处理措施               |
|-----|---------------|----------------------------------------------------------------|-----------------|--------------------|
| 400 | EndPoint.2003 | The endpoint service does not exist.                           | 终端节点服务不存在。      | 请输入终端节点服务。         |
| 400 | EndPoint.2004 | The endpoint service is unavailable.                           | 终端节点服务不可用。      | 请稍后重试，如若不行请联系技术支持。 |
| 400 | EndPoint.2009 | The specification information does not exist.                  | 规格不存在。          | 请输入正确规格。           |
| 400 | EndPoint.2010 | The input parameter subnet ID is empty.                        | 输入参数子网ID为空。     | 请输入有效的子网ID。        |
| 400 | EndPoint.2011 | The input parameter VPC ID is empty.                           | 输入参数VPC ID为空。   | 请输入有效的参数VPC ID。    |
| 400 | EndPoint.2012 | You have no permission to connect to the VPC endpoint service. | 您没有权限连接该终端节点服务。 | 请确认是否有权访问。         |
| 400 | EndPoint.2013 | The endpoint does not belong to the endpoint service.          | 终端节点不属于终端节点服务。  | 请确认终端节点是否正确。       |
| 400 | EndPoint.2014 | The endpoint has connected to the endpoint service.            | 终端节点已连接终端节点服务。  | 已经连接，无需再次连接。       |
| 400 | EndPoint.2015 | The endpoint has been frozen.                                  | 终端节点被冻结。        | 请确认冻结原因，联系技术支持。    |
| 400 | EndPoint.2016 | The endpoint pool IP address does not exist.                   | 终端节点pool IP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 | EndPoint.2017 | Invalid endpoint ID.                            | 终端节点ID无效。                 | 请输入正确的Endpoint ID。     |
| 400 | EndPoint.2018 | The endpoint is being deleted.                  | 终端节点正在删除中。                | 请选择可用的终端节点。            |
| 400 | EndPoint.2019 | The endpoint is being created.                  | 终端节点正在创建中。                | 请稍后重试。                 |
| 400 | EndPoint.2020 | qrMac or sgMac does not exist.                  | 查询qrMac或sgMac不存在。         | 请联系技术支持。               |
| 400 | EndPoint.2021 | Failed to query the VPC.                        | 查询VPC失败。                  | 请联系技术支持。               |
| 400 | EndPoint.2022 | Failed to create an endpoint.                   | 终端节点创建失败。                 | 请联系技术支持。               |
| 400 | EndPoint.2023 | CIDR is not found.                              | 终端节点未获取cidr。              | 请联系技术支持。               |
| 400 | EndPoint.2024 | shadowVpc or shadowPort does not exist.         | shadowVpc 或 shadowPort为空。 | 请联系技术支持。               |
| 400 | EndPoint.2025 | The endpoint port does not exist.               | 终端节点端口不存在。                | 请联系技术支持。               |
| 400 | EndPoint.2026 | VNI is empty.                                   | VNI为空。                    | 请联系技术支持。               |
| 400 | EndPoint.2027 | Invalid action.                                 | 操作类型不存在。                  | 请输入有效的操作类型。            |
| 400 | EndPoint.2028 | The endpoint service port or protocol is empty. | 终端节点服务端口或协议为空。            | 请输入有效的终端节点服务端口或协议。     |
| 400 | EndPoint.2029 | The requested endpoint service ID is empty.     | 被请求的终端节点服务ID为空。           | 请输入有效的终端节点服务ID。        |
| 400 | EndPoint.2031 | Only one endpoint is allowed.                   | 当前只支持单个业务。                | 接受/拒绝终端节点服务的连接只支持单个业务。 |

| 状态码 | 错误码           | 错误信息                                                      | 描述                   | 处理措施               |
|-----|---------------|-----------------------------------------------------------|----------------------|--------------------|
| 400 | EndPoint.2033 | The entered parameter enable_dns is invalid.              | 输入参数 enable_dns 无效。  | 请输入有效的 enable_dns。 |
| 400 | EndPoint.2034 | The entered parameter enable_dns is incorrect.            | 输入参数 enable_dns 错误。  | 请输入正确的 enable_dns。 |
| 400 | EndPoint.2035 | The system parameter dns.enable is invalid.               | 系统参数 dns.enable 无效。  | 请联系技术支持。           |
| 400 | EndPoint.2037 | The current network does not belong to the VPC.           | 当前 Network 不属于该 VPC。 | 请核对参数是否正确。         |
| 400 | EndPoint.2038 | The pool does not exist.                                  | 资源池不存在。              | 请联系技术支持。           |
| 400 | EndPoint.2039 | The route table is being used by another VPC endpoint.    | 路由表已被终端节点使用。         | 请联系技术支持。           |
| 400 | EndPoint.2040 | The VPC endpoint has no route table bound.                | 当前终端节点未绑定路由表。        | 请联系技术支持。           |
| 400 | EndPoint.2041 | Invalid Port IP address.                                  | 端口 IP 地址无效。          | 请输入正确的端口 IP。       |
| 400 | EndPoint.2042 | The Port IP is in use.                                    | 端口 IP 正在使用。          | 请输入正确的端口 IP。       |
| 400 | EndPoint.2043 | The Port IP is not valid for the subnet.                  | 端口 IP 对子网无效。         | 请输入正确的端口 IP。       |
| 400 | EndPoint.2044 | Invalid whitelist.                                        | 白名单无效。               | 请输入正确的参数。          |
| 400 | EndPoint.2045 | The maximum number of whitelist records has been reached. | 已达到白名单记录的最大数量。       | 请联系技术支持。           |

| 状态码 | 错误码           | 错误信息                                                      | 描述                   | 处理措施           |
|-----|---------------|-----------------------------------------------------------|----------------------|----------------|
| 400 | EndPoint.2046 | Can not modify a specified mac endpoint.                  | 无法修改指定的mac终端节点。      | 请联系技术支持。       |
| 400 | EndPoint.2047 | The VPC endpoint in the Failed status cannot be modified. | 状态为“失败”的VPC终端节点不能修改。 | 请选择有效的VPC终端节点。 |
| 400 | EndPoint.2048 | The VPC endpoint policy is invalid.                       | VPC终端节点策略无效。         | 请输入正确的参数。      |
| 400 | EndPoint.2049 | The VPC endpoint has no policy associated.                | VPC终端节点没有已关联的策略。     | 请联系技术支持。       |
| 400 | EndPoint.2050 | The endpoint has been disabled.                           | 终端节点已被禁用。            | 请联系技术支持。       |
| 400 | EndPoint.2051 | The current endpoint is switching to new pool.            | 终端端点正在切换到新资源池。       | 请稍后重试。         |
| 400 | EndPoint.2052 | The current endpoint does not need to be rolled back.     | 终端节点无法回退。            | 请联系技术支持。       |
| 400 | EndPoint.3001 | Failed to create a port.                                  | 创建端口失败。              | 请联系技术支持。       |
| 400 | EndPoint.3002 | Invalid permission.                                       | 权限无效。                | 请联系技术支持。       |
| 400 | EndPoint.3003 | Invalid port ID.                                          | 端口ID无效。              | 请联系技术支持。       |
| 400 | EndPoint.3004 | Invalid port.                                             | 无效的port。             | 请联系技术支持。       |
| 400 | EndPoint.3005 | Failed to delete the endpoint service.                    | 无法删除终端节点服务。          | 请联系技术支持。       |

| 状态码 | 错误码           | 错误信息                                                                    | 描述                              | 处理措施               |
|-----|---------------|-------------------------------------------------------------------------|---------------------------------|--------------------|
| 400 | EndPoint.3006 | The endpoint service is being used.                                     | 终端节点服务被终端节点使用。                  | 请联系技术支持。           |
| 400 | EndPoint.3008 | The port does not exist.                                                | 端口不存在。                          | 请联系技术支持。           |
| 400 | EndPoint.3009 | Invalid CIDR.                                                           | cidr无效。                         | 请联系技术支持。           |
| 400 | EndPoint.3010 | Invalid IP address.                                                     | IP无效。                           | 请输入正确的IP。          |
| 400 | EndPoint.3011 | Parameter IP is not required to create an endpoint service (interface). | 在创建 interface类型终端节点服务时，不需要参数IP。 | 请输入正确的请求体。         |
| 400 | EndPoint.3013 | endpointService interface vlan can't have vpcId.                        | vlan类型的终端节点服务请求不能包含 vpcId。      | 请输入正确的请求体。         |
| 400 | EndPoint.3014 | endpointService interface can't have cidr.                              | interface类型终端节点服务请求不能包含 cidr。   | 请输入正确的请求体。         |
| 400 | EndPoint.3015 | endpointService gateway vlan can't have portId.                         | vlan类型的终端节点服务请求不能包含 portId。     | 请输入正确的请求体。         |
| 400 | EndPoint.3016 | endpointService gateway vlan can't have ip.                             | vlan类型的终端节点服务请求不能包含 ip。         | 请输入正确的请求体。         |
| 400 | EndPoint.3017 | Invalid CIDRs.                                                          | 无效的 CIDRs。                      | 请输入正确的 CIDRs参数。    |
| 400 | EndPoint.3018 | endpointService gateway vlan can't have vpcId.                          | vlan类型的终端节点服务请求不能包含 vpcId。      | 请输入正确的请求体。         |
| 400 | EndPoint.3021 | Invalid serverType.                                                     | serverType参数无效。                 | 请输入有效的 serverType。 |
| 400 | EndPoint.3022 | Failed to create a network.                                             | 创建网络失败。                         | 请联系技术支持。           |

| 状态码 | 错误码           | 错误信息                                                                       | 描述                                | 处理措施           |
|-----|---------------|----------------------------------------------------------------------------|-----------------------------------|----------------|
| 400 | EndPoint.3023 | Failed to create a subnet.                                                 | 创建子网失败。                           | 请联系技术支持。       |
| 400 | EndPoint.3035 | Invalid action.                                                            | 终端节点服务授权操作错误。                     | 请输入正确的授权操作。    |
| 400 | EndPoint.3036 | Invalid permissions.                                                       | 权限列表不能为空。                         | 请输入正确的请求体。     |
| 400 | EndPoint.3040 | Failed to add a rollback task.                                             | 添加回滚任务失败。                         | 请联系技术支持。       |
| 400 | EndPoint.3042 | The port ID does not belong to the current VPC.                            | 端口ID不属于当前VPC。                     | 请输入正确的请求体。     |
| 400 | EndPoint.3043 | The service port is invalid.                                               | 服务端口无效。                           | 请输入正确的请求体。     |
| 400 | EndPoint.3044 | The parameter ports conflicted with ports in an existing endpoint service. | Ports与已有endpoint service的Ports冲突。 | 请输入正确的请求体。     |
| 400 | EndPoint.3045 | Other properties cannot be modified in the current endpoint service state. | 终端节点服务目前的状态不支持修改其他属性。             | 请输入正确的请求体。     |
| 400 | EndPoint.3046 | The IP address conflicted with an existing endpoint service.               | IP和已有的终端节点服务冲突。                   | 请输入正确的请求体。     |
| 400 | EndPoint.3048 | Invalid netType.                                                           | netType无效。                        | 请输入有效的netType。 |

| 状态码 | 错误码           | 错误信息                                                                             | 描述                     | 处理措施                |
|-----|---------------|----------------------------------------------------------------------------------|------------------------|---------------------|
| 400 | EndPoint.3049 | The maximum number of whitelist records has been reached.                        | 终端节点服务白名单记录总量超过限制。     | 删除无效白名单或将白名单权限设置为*。 |
| 400 | EndPoint.3051 | Endpoint service vip port id is invalid.                                         | vip port id无效。         | 请输入正确的vip port id。  |
| 400 | EndPoint.3052 | portId and ip cannot be modified at the same time.                               | portId和ip不能同时修改。       | 请输入正确的请求体。          |
| 400 | EndPoint.3053 | vipPortId and ip cannot be modified at the same time.                            | vipPortId和ip不能同时修改。    | 请输入正确的请求体。          |
| 400 | EndPoint.3054 | portId or vipPortId cannot be modified.                                          | 不能修改portId或vipPortId。  | 请输入正确的请求体。          |
| 400 | EndPoint.3055 | ip cannot be modified.                                                           | 不能修改ip。                | 请输入正确的请求体。          |
| 400 | EndPoint.3056 | The maximum of VPC endpoint services using the same IP address has been reached. | 单个后端资源支持的终端节点服务数量超过限制。 | 请联系技术支持。            |
| 400 | EndPoint.3057 | cidr cannot be modified.                                                         | 不能修改cidr。              | 请输入正确的请求体。          |
| 400 | EndPoint.3058 | The domain name is invalid.                                                      | dns name不合法。           | 请输入正确的dns name参数。   |
| 400 | EndPoint.3059 | The domain name already exists.                                                  | dns name已经存在。          | 请联系技术支持。            |



| 状态码 | 错误码           | 错误信息                                                   | 描述               | 处理措施                          |
|-----|---------------|--------------------------------------------------------|------------------|-------------------------------|
| 400 | EndPoint.3060 | You have no permission to add domain names.            | 没有权限添加 dns name。 | 请联系技术支持。                      |
| 400 | EndPoint.3061 | The maximum number of domain names has reached.        | dns name个数超过限制。  | 请联系技术支持。                      |
| 400 | EndPoint.3062 | Invalid endpoint service ID.                           | 终端节点服务 ID无效。     | 请输入正确的 endpoint_service_id参数。 |
| 400 | EndPoint.3063 | Invalid port ID.                                       | 端口ID无效。          | 请输入正确的端口 ID。                  |
| 400 | EndPoint.3066 | The tag cannot be empty.                               | 标签不能为空。          | 请输入正确的请求体。                    |
| 400 | EndPoint.3067 | The tag key cannot be duplicated.                      | 标签键不能重复。         | 请输入正确的请求体。                    |
| 400 | EndPoint.3068 | Tag keys and values should meet relevant requirements. | 标签键和值必须符合规范。     | 请输入正确的请求体。                    |
| 400 | EndPoint.3069 | The maximum number of tags has been reached.           | 标签总数超出限制。        | 请联系技术支持。                      |
| 400 | EndPoint.3070 | Invalid resource type.                                 | 资源类型错误。          | 请联系技术支持。                      |
| 400 | EndPoint.3071 | The tag value cannot be duplicated.                    | 标签值不能重复。         | 请联系技术支持。                      |
| 400 | EndPoint.3072 | The tag key size is invalid.                           | 标签键长度不符合规范。      | 请输入规范的标签键。                    |
| 400 | EndPoint.3073 | The tag value size is invalid.                         | 标签值长度不符合规范。      | 请输入规范的标签值。                    |

| 状态码 | 错误码           | 错误信息                                   | 描述                 | 处理措施                 |
|-----|---------------|----------------------------------------|--------------------|----------------------|
| 400 | EndPoint.3074 | The maximum of ports has been reached. | 端口映射个数超出限制。        | 请联系技术支持。             |
| 400 | EndPoint.3075 | The protocol is invalid.               | 协议无效。              | 请联系技术支持。             |
| 400 | EndPoint.3076 | Invalid service name.                  | service name 是非法的。 | 请输入合法的 service name。 |
| 400 | EndPoint.4001 | Failed to query the subnet.            | 查询子网失败。            | 请联系技术支持。             |
| 400 | EndPoint.4002 | Failed to create a subnet.             | 创建子网失败。            | 请联系技术支持。             |
| 400 | EndPoint.4003 | Failed to delete the subnet.           | 删除子网失败。            | 请联系技术支持。             |
| 400 | EndPoint.4005 | Failed to query the network.           | 查询network失败。       | 请联系技术支持。             |
| 400 | EndPoint.4006 | Failed to create a network.            | 创建network失败。       | 请联系技术支持。             |
| 400 | EndPoint.4007 | Failed to delete the network.          | 删除network失败。       | 请联系技术支持。             |
| 400 | EndPoint.4009 | Failed to query the port.              | 查询端口失败。            | 请联系技术支持。             |
| 400 | EndPoint.4010 | Failed to create a port.               | 创建端口失败。            | 请联系技术支持。             |
| 400 | EndPoint.4011 | Failed to delete the port.             | 删除端口失败。            | 请联系技术支持。             |
| 400 | EndPoint.4013 | Failed to query the proxy.             | 查询代理失败。            | 请联系技术支持。             |
| 400 | EndPoint.4014 | Failed to query the router.            | 查询路由失败。            | 请联系技术支持。             |

| 状态码 | 错误码           | 错误信息                                                | 描述               | 处理措施     |
|-----|---------------|-----------------------------------------------------|------------------|----------|
| 400 | EndPoint.4015 | The router is not found.                            | 未找到路由。           | 请联系技术支持。 |
| 400 | EndPoint.4016 | Failed to add an interface router.                  | 添加interface路由失败。 | 请联系技术支持。 |
| 400 | EndPoint.4017 | Failed to delete the interface router.              | 删除interface路由失败。 | 请联系技术支持。 |
| 400 | EndPoint.4018 | Failed to add an extension router.                  | 添加扩展路由失败。        | 请联系技术支持。 |
| 400 | EndPoint.4019 | Failed to delete the extension router.              | 删除扩展路由失败。        | 请联系技术支持。 |
| 400 | EndPoint.4020 | Failed to query Neutron L3 Agent.                   | 查询路由L3-agent失败。  | 请联系技术支持。 |
| 400 | EndPoint.4025 | The specification is being used.                    | 规格已经被使用。         | 请联系技术支持。 |
| 400 | EndPoint.4026 | Failed to query the default route table of the VPC. | 查询VPC默认路由表失败。    | 请联系技术支持。 |
| 400 | EndPoint.4027 | Failed to query route tables of the VPC.            | 查询VPC路由表列表失败。    | 请联系技术支持。 |
| 400 | EndPoint.4028 | Failed to add routes to the VPC's route table.      | 添加路由到VPC路由表失败。   | 请联系技术支持。 |
| 400 | EndPoint.4029 | Failed to remove routes from the VPC's route table. | 从VPC路由表中删除路由失败。  | 请联系技术支持。 |

| 状态码 | 错误码           | 错误信息                                                                | 描述              | 处理措施                         |
|-----|---------------|---------------------------------------------------------------------|-----------------|------------------------------|
| 401 | EndPoint.0003 | Authentication failed or authentication information is invalid.     | 鉴权失败或未提供有效鉴权信息。 | 确认是否开通权限。                    |
| 403 | EndPoint.0004 | Authentication information is incorrect or you have no permissions. | 鉴权信息错误或无相应权限。   | 确认是否开通权限。                    |
| 404 | EndPoint.0005 | The requested resource is unavailable.                              | 请求的相关资源不存在。     | 确认输入的参数是否正确。                 |
| 404 | EndPoint.2006 | The requested endpoint does not exist.                              | 请求的终端节点不存在。     | 请输入正确的终端节点。                  |
| 404 | EndPoint.2007 | The endpoint information does not exist.                            | 终端节点信息不存在。      | 请输入正确的终端节点，确认终端节点是否被删除。      |
| 404 | EndPoint.2008 | The endpoint has been deleted.                                      | 终端节点已经被删除。      | 确认终端节点是否被删除。                 |
| 404 | EndPoint.2030 | markerId is empty.                                                  | 查询markerId为空。   | 请输入有效的marker ID。             |
| 404 | EndPoint.4004 | The subnet is not found.                                            | 子网未找到。          | 请核对输入的subnet_id，如若不行请联系技术支持。 |
| 404 | EndPoint.4008 | Network is unavailable.                                             | 获取不到network。    | 请联系技术支持。                     |
| 404 | EndPoint.4012 | The port is not found.                                              | 端口未找到。          | 请联系技术支持。                     |
| 404 | EndPoint.4021 | Neutron L3 Agent is not found.                                      | 路由L3-agent不存在。  | 请联系技术支持。                     |
| 404 | EndPoint.4030 | The route table is not found.                                       | 未找到路由表。         | 请联系技术支持。                     |
| 500 | EndPoint.0001 | System error. Please retry.                                         | 系统异常，请重试。       | 请重试，如若再次出现异常，请联系技术支持。        |

| 状态码 | 错误码           | 错误信息                        | 描述      | 处理措施         |
|-----|---------------|-----------------------------|---------|--------------|
| 504 | EndPoint.0011 | The request body is null.   | 请求体为空。  | 请重新输入请求体。    |
| 504 | EndPoint.0012 | The request header is null. | 请求体头为空。 | 请输入请求头。      |
| 504 | EndPoint.0013 | The request timed out.      | 请求超时。   | 重试失败请联系技术支持。 |

## 6.3 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": "65ewtrgaggshhk1223245sghjlse684b",
 "is_domain": false,
 "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
 "name": "project_name",
 "description": "",
 "links": {
 "next": null,
 "previous": null,
 "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
 },
 "id": "a4adasfjljaaaakla12334jklga9sasfg",
 "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 6-1 查看项目 ID

