VPC Endpoint

API Reference

Issue 01

Date 2024-09-11

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

1.1 Overview

Welcome to *VPC Endpoint API Reference*. The VPC Endpoint service provides secure and private channels to connect your VPCs to VPC endpoint services (cloud services on the current platform or your private services), providing flexible networking without having to use EIPs.

This document describes how to use application programming interfaces (APIs) to perform operations on VPC Endpoint resources, such as creating, querying, modifying, or deleting VPC endpoints and VPC endpoint services. For details about all supported operations, see **API Overview**.

If you plan to access VPC Endpoint through an API, ensure that you are familiar with VPC Endpoint concepts. For details, see **Service Overview**.

1.2 API Calling

VPCEP supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints for accessing the VPCEP service, see **Regions and Endpoints**.

1.4 Constraints

- The number of VPCEP resources that you can create is determined by your quota. To view or increase the quota, see **Quota Adjustment**.
- For detailed constraints, see the constraints described in specific APIs.

1.5 Concepts

- Account

  An account is created upon successful signing up. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

  An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  API authentication requires information such as the account name, username, and password.

- Region

  Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

  For details, see **Region and AZ**.

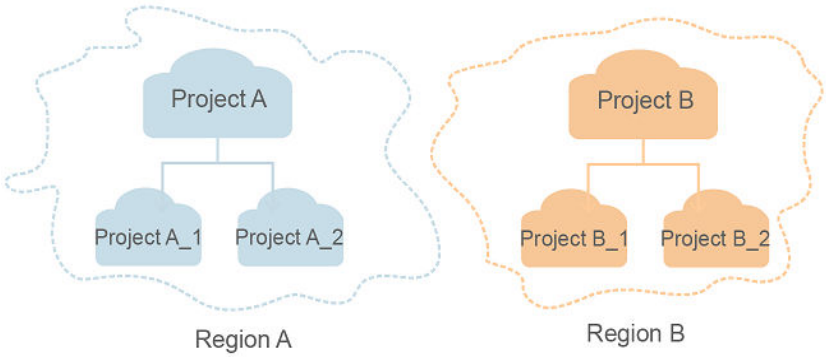
- AZ

  An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

  A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



- Enterprise Project

  Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

  For details about enterprise projects and about how to obtain enterprise project IDs, see ***Enterprise Management User Guide***.

2 API Overview

The VPCEP service provides extended RESTful APIs.

VPCEP APIs allow you to use all VPCEP functions. VPCEP has two types of resources: VPC endpoints and VPC endpoint services.

Table 2-1 describes the APIs provided by VPCEP.

Table 2-1 API overview

API	Description
Version management APIs	APIs for querying version information of all VPCEP APIs or a specified API
VPC endpoint service APIs	• APIs for creating, querying, modifying, or deleting a VPC endpoint service • API for querying the list of VPC endpoint services • APIs for adding, querying, or deleting a whitelist record • APIs for the endpoints connected to and those accepted or rejected to connect to a VPC endpoint service With these APIs, you can manage VPC endpoint services and set rules based on service conditions to provide services for VPC endpoints.
VPC endpoint APIs	• APIs for creating, querying, deleting a VPC endpoint • API for querying the list of VPC endpoints With these APIs, you can manage VPC endpoints and use services provided by VPC endpoint services.
Resource quota API	API for querying the quota of VPCEP resources
Tag API	API for managing VPCEP tags, including querying resources by tag, adding and deleting a tag or tags, and querying resource tags

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **creating an IAM User** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

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

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in region **CN-Hong Kong** is **iam.ap-southeast-1.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of *Parameter name*=*Parameter value*. For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

IAM is a global service. You can create an IAM user using the endpoint of IAM in any region. For example, to create an IAM user in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3.0/OS-USER/users**) in the URI of the API for **creating an IAM user**. Then construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Figure 3-1 Example URI

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.

Method	Description
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API for **creating an IAM user**, the request method is **POST**. An example request is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for **obtaining a user token** (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

NOTE

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in **Authentication**.

The following shows an example request of the API for **creating an IAM user** when AK/SK authentication is used:

```
POST https://iam.ap-southeast-1.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=**********
```

(Optional) Request Body

This part is optional. A request body is generally sent in a structured format (for example, JSON or XML), which is specified by **Content-Type** in the request header. It is used to transfer content other than the request header. If the request body contains full-width characters, these characters must be coded in UTF-8.

The request body varies depending on APIs. Certain APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

The following shows an example request (a request body included) of the API for **creating an IAM user**. You can learn about request parameters and related

description from this example. The bold parameters need to be replaced for a real request.

- **accountid**: account ID of an IAM user
- **username**: name of an IAM user
- **email**: email of an IAM user
- **password**: login password of an IAM user

```
POST https://iam.ap-southeast-1.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"
    }
}
```

If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **X-Subject-Token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.
- Token authentication: Requests are authenticated using tokens.

AK/SK Authentication

NOTE

AK/SK authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token authentication is recommended.

In AK/SK authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see **API Request Signing Guide**.

NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

Token Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API. You can obtain a token by calling the **Obtaining User Token** API.

IMS is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",   // IAM user name
                    "password": $ADMIN_PASS,  //IAM user password. You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "domain": {
                        "name": "domainname"  // Name of the account to which the IAM user belongs
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"    // Project name
            }
        }
    }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status Code**.

For example, if status code **201** is returned for calling the API used to **create an IAM user**, the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-2 shows the response header fields for the API used to **create an IAM user**. The **X-Subject-Token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-2 Header fields of the response to the request for creating an IAM user


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


(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to **create an IAM user**.


```
{
    "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
```



```
    }
}
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_msg": "The request message format is invalid.",
    "error_code": "IMG.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 API

4.1 Version Management

4.1.1 Querying Versions of VPC Endpoint APIs

Function

This API is used to query versions of VPC Endpoint APIs.

Calling Method

For details, see **Calling APIs**.

URI

GET /

Request Parameters

Table 4-1 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-2 Response body parameters

Parameter	Type	Description
versions	Array of **VersionObject** objects	Specifies the list of VPCEP versions.

Table 4-3 VersionObject

Parameter	Type	Description
status	String	Specifies the version status.● CURRENT: indicates a primary version.● SUPPORT: indicates an earlier version which is still supported. ● DEPRECATED: indicates deprecated version which may be deleted later.
id	String	Specifies the API version ID. Minimum: **1** Maximum: **16**
updated	String	Specifies the time the API version was released. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
version	String	Specifies the supported version number.
min_version	String	Specifies the supported microversion number. If the APIs do not support microversions, the parameter is left blank.
links	Array of **Link** objects	Specifies the API URL.

Table 4-4 Link

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
type	String	Specifies the MIME type of the request body. The value is application/json.
rel	String	Specifies the relationship between the current API version and the referenced address.

Example Requests

Querying versions of VPCEP APIs

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.1.2 Querying the Version of a VPC Endpoint API

Function

This API is used to query the version of a VPC Endpoint API.

Calling Method

For details, see **Calling APIs**.

URI

GET /{version}

Table 4-5 Path Parameters

Parameter	Mandatory	Type	Description
version	Yes	String	Specifies the version number to be queried. The value starts with v, for example, v1. If this parameter is left blank, versions of all VPCEP APIs are queried.

Request Parameters

Table 4-6 Request header parameters

Parameter	Mandatory	Type	Description
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-7 Response body parameters

Parameter	Type	Description
version	**VersionObject** object	Specifies the VPCEP API versions.

Table 4-8 VersionObject

Parameter	Type	Description
status	String	Specifies the version status.● CURRENT: indicates a primary version.● SUPPORT: indicates an earlier version which is still supported. ● DEPRECATED: indicates deprecated version which may be deleted later.

Parameter	Type	Description
id	String	Specifies the API version ID. Minimum: **1** Maximum: **16**
updated	String	Specifies the time the API version was released. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
version	String	Specifies the supported version number.
min_version	String	Specifies the supported microversion number. If the APIs do not support microversions, the parameter is left blank.
links	Array of **Link** objects	Specifies the API URL.

Table 4-9 Link

Parameter	Type	Description
href	String	Specifies the reference address of the current API version.
type	String	Specifies the MIME type of the request body. The value is application/json.
rel	String	Specifies the relationship between the current API version and the referenced address.

Example Requests

This API is used to query the v1 version of a specified VPCEP API.

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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();
        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.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 = ListSpecifiedVersionDetailsRequest()
        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{}
    response, err := client.ListSpecifiedVersionDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```



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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2 VPC Endpoint Services

4.2.1 Creating a VPC Endpoint Service

Function

This API is used to create a VPC endpoint service. Other users can create a VPC endpoint to connect to the VPC endpoint service.

Note: This API is asynchronous. If it is successfully invoked, status code **200** is returned, indicating that the request has been successfully delivered. It takes 1 to 2 minutes for the system to provision a VPC endpoint service. You can view the creation result by querying the details about the VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-10 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-11 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It is a response to the API for obtaining a user token. This API is the only one that does not require authentication.The value of X-Subject-Token in the response header is the token value.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-12 Request body parameters

Parameter	Mandatory	Type	Description
port_id	Yes	String	Specifies the ID that identifies the backend resource of the VPC endpoint service. The ID is in UUID format. The value can be: • The port ID of the private IP address of a load balancer (recommended). For details, see section "Viewing Details of a Load Balancer" in the *Elastic Load Balance API Reference*. • The NIC ID of an ECS IP address. For details, see the **port_id** field in the response parameters of section "Querying NICs of an ECS" in the *Elastic Cloud Server API Reference*. • The NIC ID of the virtual server for which the virtual IP address is configured (discarded). Note: • You cannot create a VPC endpoint service in a VPC whose CIDR block overlaps with 198.19.128.0/17. • The destination address of the custom route in the VPC route table cannot overlap with 198.19.128.0/17. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
service_name	No	String	Specifies the name of the VPC endpoint service. The name can contain a maximum of 16 characters, including letters, digits, underscores (_), and hyphens (-). ● If you do not specify this parameter, the VPC endpoint service name is in **regionName.serviceId** format: ● If you specify a valid value, the VPC endpoint service name is in the following format: **regionNameserviceNameserviceId**. Minimum: **0** Maximum: **16**
vpc_id	Yes	String	Specifies the ID of the VPC where the backend resource of the VPC endpoint service is located. For details, see the id field in the response information of the section "Querying VPC Details" in the Virtual Private Cloud API Reference. Minimum: **1** Maximum: **64**
approval_enabled	No	Boolean	Specifies whether approval is required. ● false: No approval is required. The created VPC endpoint is in the accepted state by default. ● true: Approval is required. The created VPC endpoint is in the pendingAcceptance state and can be used only after being approved by the user of the VPC endpoint service. The default value is true. Default: **true**

Parameter	Mandatory	Type	Description
service_type	No	String	Specifies the type of the VPC endpoint service. Only your private services can be configured into interface VPC endpoint services. There are two types of VPC endpoint services: interface and gateway. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You can directly use the cloud services configured by the O&M personnel without creating them by yourself. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create VPC endpoints to connect to gateway and interface VPC endpoint services. Default: **interface**
server_type	Yes	String	Specifies the resource type. • **VM**: indicates a cloud server. • **VIP**: indicates a virtual IP address. (This value has been discarded. **LB** is recommended.) • **LB**: indicates a load balancer that works well for high-traffic services that require high reliability and disaster recovery (DR) performance.

Parameter	Mandatory	Type	Description
ip	No	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.
ports	Yes	Array of **PortList** objects	Specifies the port mappings opened by the VPC endpoint service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share one port ID, the combinations of server ports and protocols for all port mappings between VPC endpoint services must be unique. A maximum of 200 port mappings can be added at a time.

Parameter	Mandatory	Type	Description
tcp_proxy	No	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**. Default: **close**
tags	No	Array of **TagList** objects	Resource tag list A maximum of 10 tags can be added to a VPC endpoint service.

Parameter	Mandatory	Type	Description
description	No	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) and great than signs (>). Minimum: **0** Maximum: **512**
ip_version	No	String	Specifies the IP version of the VPC endpoint service. Only professional VPC endpoint services support this parameter. ● **ipv4**: The IP address of the VPC endpoint service is an IPv4 address. ● **ipv6**: The IP address of the VPC endpoint service is an IPv6 address.

Table 4-13 PortList

Parameter	Mandatory	Type	Description
client_port	No	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	No	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**

Parameter	Mandatory	Type	Description
protocol	No	String	Port mapping protocol. TCP is supported. Default: **TCP**

Table 4-14 TagList

Parameter	Mandatory	Type	Description
key	No	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	No	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Response Parameters

Status code: 200

Table 4-15 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
port_id	String	Specifies the ID that identifies the backend resource of the VPC endpoint service. The ID is in UUID format. The value can be: • **LB**: indicates the port ID of the private IP address of a load balancer (recommended). • **VM**: indicates the NIC ID of an ECS IP address. • **VIP**: indicates the NIC ID of the virtual server for which the virtual IP address is configured. (This value has been discarded. **LB** is recommended.) Minimum: **1** Maximum: **64**
service_name	String	Specifies the name of the VPC endpoint service. Minimum: **0** Maximum: **128**
server_type	String	Specifies the resource type. • **VM**: cloud server • **VIP**: virtual IP address • **LB**: enhanced load balancer
vpc_id	String	Specifies the ID of the VPC where the backend resource of the VPC endpoint service is located. Minimum: **1** Maximum: **64**
pool_id	String	Specifies the cluster ID of the VPC endpoint service.

Parameter	Type	Description
approval_enabled	Boolean	Specifies whether connection approval is required. • **false**: Connection approval is not required. The created VPC endpoint is in the **accepted** state. • **true**: Connection approval is required. The created VPC endpoint is in the **pendingAcceptance** state, and it can be used only after being approved by the user of the VPC endpoint service.
status	String	Specifies the status of the VPC endpoint service. • **creating**: The VPC endpoint service is being created. • **available**: The VPC endpoint service is connectable. • **failed**: The VPC endpoint service failed to be created.
service_type	String	Specifies the type of the VPC endpoint service. There are two types of VPC endpoint services: interface and gateway. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can create VPC endpoints to connect to gateway and interface VPC endpoint services. Minimum: **1** Maximum: **16**
created_at	String	Specifies when the VPC endpoint service was created. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
updated_at	String	Specifies when the VPC endpoint service was updated. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
project_id	String	Specifies the project ID. Minimum: **1** Maximum: **64**

Parameter	Type	Description
ip	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.
ports	Array of **PortList** objects	Specifies the port mappings opened to the VPC endpoint service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share one **port_id**, either **server_port** or **protocol**, or both **server_port** and **protocol** of each of these endpoint services must be unique.
tcp_proxy	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**.
tags	Array of **TagList** objects	Specifies resource tags.
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**

Parameter	Type	Description
enable_policy	Boolean	Specifies whether the VPC endpoint policy can be customized. • **false**: The VPC endpoint policy cannot be customized. • **true**: The VPC endpoint policy can be customized. The default value is **false**.
ip_version	String	Specifies the IP version of the VPC endpoint service. Only professional VPC endpoint services support this parameter. • **ipv4**: The IP address of the VPC endpoint service is an IPv4 address. • **ipv6**: The IP address of the VPC endpoint service is an IPv6 address.

Table 4-16 PortList

Parameter	Type	Description
client_port	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
protocol	String	Port mapping protocol. TCP is supported. Default: **TCP**

Table 4-17 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Example Requests

Creating an interface VPC endpoint service (Setting **approval_enabled** to **false**, **service type** to **VM**, **client_port** to **8080** and **8081**, **server_port** to **90** and **80**, and **protocol** to **TCP**)

```
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"
  } ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

Creating an interface VPC endpoint service (Setting **approval_enabled** to **false**, **service type** to **VM**, **client_port** to **8080** and **8081**, **server_port** to **90** and **80**, and **protocol** to **TCP**)

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

        ICredential auth = new BasicCredentials()
                .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.withVpcId("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

Creating an interface VPC endpoint service (Setting **approval_enabled** to **false**, **service type** to **VM**, **client_port** to **8080** and **8081**, **server_port** to **90** and **80**, and **protocol** to **TCP**)

```
# 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 = 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

Creating an interface VPC endpoint service (Setting **approval_enabled** to **false**, **service type** to **VM**, **client_port** to **8080** and **8081**, **server_port** to **90** and **80**, and **protocol** to **TCP**)

```
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.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,
        VpcId: "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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.2 Querying VPC Endpoint Services

Function

This API is used to query VPC endpoint services.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-18 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID Minimum: **1** Maximum: **64**

Table 4-19 Query Parameters

Parameter	Mandatory	Type	Description
endpoint_service_name	No	String	Specifies the name of the VPC endpoint service. The name is not case-sensitive and supports fuzzy match. Minimum: **1** Maximum: **128**
id	No	String	Specifies the unique ID of the VPC endpoint service.
status	No	String	Specifies the status of the VPC endpoint service. ● creating: The VPC endpoint service is being created. ● available: The VPC endpoint service can be connected. ● failed: The VPC endpoint service fails to be created. ● deleting: The VPC endpoint service is being deleted.
sort_key	No	String	Specifies the sorting field of the VPC endpoint services, which can be: ● create_at: VPC endpoint services are sorted by creation time. ● update_at: VPC endpoint services are sorted by update time. The default field is create_at. Default: **create_at**

Parameter	Mandatory	Type	Description
sort_dir	No	String	Specifies the sorting method of VPC endpoint services, which can be: ● desc: VPC endpoint services are sorted in descending order. ● asc: VPC endpoint services are sorted in ascending order. The default method is desc. Default: **desc**
limit	No	Integer	Specifies the maximum number of VPC endpoint services displayed on each page. The value ranges from 0 to 1000 and is generally 10, 20, or 50. The default number is 10. Minimum: **1** Maximum: **1000** Default: **10**
offset	No	Integer	Specifies the offset. All VPC endpoint services after this offset will be queried. The offset must be an integer greater than 0 but less than the number of VPC endpoint services. Minimum: **0**
public_border_group	No	String	Specifies the VPC endpoint service that matches the edge attribute in the filtering result.
net_type	No	String	Specifies the type of the network to which the IPv4 address of the VPC endpoint belongs. Minimum: **1** Maximum: **5**

Request Parameters

Table 4-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-21 Response body parameters

Parameter	Type	Description
endpoint_services	Array of **ServiceList** objects	Specifies VPC endpoint services.
total_count	Integer	Specifies the total number of VPC endpoint services that meet the search criteria. The number is not affected by **limit** or **offset**.

Table 4-22 ServiceList

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Parameter	Type	Description
port_id	String	Specifies the ID that identifies the backend resource of the VPC endpoint service. The ID is in UUID format. If the backend resource is ● A load balancer (recommended), specify the port ID of the private IP address of the load balancer. ● An ECS, specify the NIC ID of the ECS IP address. ● A virtual IP address (discarded), specify the NIC ID of the physical server where virtual resources are created.
service_name	String	Specifies the name of the VPC endpoint service. Minimum: **0** Maximum: **16**
server_type	String	Specifies the resource type. ● VM: indicates a cloud server. ● VIP: indicates a virtual IP address. ● LB: indicates a shared load balancer.
vpc_id	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs. Minimum: **1** Maximum: **64**
approval_enabled	Boolean	Specifies whether approval is required. ● false: No approval is required. The created VPC endpoint is in the accepted state by default. ● true: Approval is required. The created VPC endpoint is in the pendingAcceptance state, and can be used only after being approved by the user of the VPC endpoint service.
status	String	Specifies the status of the VPC endpoint service. ● creating: The VPC endpoint service is being created. ● available: The VPC endpoint service can be connected. ● failed: The VPC endpoint service fails to be created. ● deleting: The VPC endpoint service is being deleted.

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service. There are two types of VPC endpoint services: interface and gateway. ● gateway: VPC endpoint services of this type are configured by O&M personnel. You can use them directly without creating them by yourselves. ● interface: Cloud services configured by O&M personnel and private services created by yourselves are included. Cloud services configured by O&M personnel do not need to be created. You can use it directly. You can create a VPC endpoint for accessing gateway and interface VPC endpoint services.
created_at	String	Specifies the creation time of the VPC endpoint service. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint service. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
project_id	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID. Minimum: **1** Maximum: **64**
domain_id	String	Domain ID Minimum: **1** Maximum: **64**
ports	Array of **PortList** objects	Specifies the port mappings opened by the VPC endpoint service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share one port ID,the combinations of server ports and protocols for all port mappings between VPC endpoint services must be unique.
ip	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.
tags	Array of **TagList** objects	Resource tag list
connection_count	Integer	Specifies the number of VPC endpoints that are in the Creating or Accepted status. Minimum: **0**

Parameter	Type	Description
tcp_proxy	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**.
error	Array of **Error** objects	Specifies the error message returned when a task submission exception occurs.
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) and great than signs (>). Minimum: **0** Maximum: **512**
public_border_group	String	Specifies the information about Public Border Group of the pool corresponding to the VPC endpoint service.
enable_policy	Boolean	Specifies whether the VPC endpoint policy is enabled. • **false**: The VPC endpoint policy cannot be set. • **true**: The VPC endpoint policy can be set. The default value is **false**.

Table 4-23 PortList

Parameter	Type	Description
client_port	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
protocol	String	Port mapping protocol. TCP is supported. Default: **TCP**

Table 4-24 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-25 Error

Parameter	Type	Description
message	String	Task error description Minimum: **0** Maximum: **1024**
code	String	Error code of a task exception. Minimum: **0** Maximum: **10**

Example Requests

Querying VPC endpoint services

```
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={limit}&offset={offset}&status={status}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        ListEndpointServiceRequest request = new ListEndpointServiceRequest();
        request.withEndpointServiceName("<endpoint_service_name>");
        request.withId("<id>");
        request.withStatus(ListEndpointServiceRequest.StatusEnum.fromValue("<status>"));
        request.withSortKey(ListEndpointServiceRequest.SortKeyEnum.fromValue("<sort_key>"));
        request.withSortDir(ListEndpointServiceRequest.SortDirEnum.fromValue("<sort_dir>"));
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withPublicBorderGroup("<public_border_group>");
        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.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 = ListEndpointServiceRequest()
        request.endpoint_service_name = "<endpoint_service_name>"
        request.id = "<id>"
        request.status = "<status>"
        request.sort_key = "<sort_key>"
        request.sort_dir = "<sort_dir>"
        request.limit = <limit>
        request.offset = <offset>
        request.public_border_group = "<public_border_group>"
        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")

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

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

    request := &model.ListEndpointServiceRequest{}
    endpointServiceNameRequest:= "<endpoint_service_name>"
    request.EndpointServiceName = &endpointServiceNameRequest
```



```
    idRequest:= "<id>"
    request.Id = &idRequest
    statusRequest:= model.GetListEndpointServiceRequestStatusEnum().<STATUS>
    request.Status = &statusRequest
    sortKeyRequest:= model.GetListEndpointServiceRequestSortKeyEnum().<SORT_KEY>
    request.SortKey = &sortKeyRequest
    sortDirRequest:= model.GetListEndpointServiceRequestSortDirEnum().<SORT_DIR>
    request.SortDir = &sortDirRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    publicBorderGroupRequest:= "<public_border_group>"
    request.PublicBorderGroup = &publicBorderGroupRequest
    response, err := client.ListEndpointService(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.3 Querying Details of a VPC Endpoint Service

Function

This API is used to query details of a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-26 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-27 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-28 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Parameter	Type	Description
port_id	String	Specifies the ID that identifies the backend resource of the VPC endpoint service. The ID is in UUID format. The value can be: • **LB**: indicates the port ID of the private IP address of a load balancer (recommended). • **VM**: indicates the NIC ID of an ECS IP address. • **VIP**: indicates the NIC ID of the virtual server for which the virtual IP address is configured. (This value has been discarded. **LB** is recommended.)
service_name	String	Specifies the name of the VPC endpoint service. Minimum: **0** Maximum: **128**
server_type	String	Specifies the resource type. • **VM**: cloud server • **VIP**: virtual IP address • **LB**: enhanced load balancer
vpc_id	String	Specifies the ID of the VPC where the backend resource of the VPC endpoint service is located. Minimum: **1** Maximum: **64**
approval_enabled	Boolean	Specifies whether connection approval is required. • **false**: Connection approval is not required. The created VPC endpoint is in the **accepted** state. • **true**: Connection approval is required. The created VPC endpoint is in the **pendingAcceptance** state, and it can be used only after being approved by the user of the VPC endpoint service.

Parameter	Type	Description
status	String	Specifies the status of the VPC endpoint service. • **creating**: The VPC endpoint service is being created. • **available**: The VPC endpoint service is connectable. • **failed**: The VPC endpoint service failed to be created. • **deleting**: The VPC endpoint service is being deleted.
service_type	String	Specifies the type of the VPC endpoint service. There are two types of VPC endpoint services: interface and gateway. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can create VPC endpoints to connect to gateway and interface VPC endpoint services.
created_at	String	Specifies when the VPC endpoint service was created. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
updated_at	String	Specifies when the VPC endpoint service was updated. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
cidr_type	String	Specifies the CIDR block type. • **public**: indicates a public CIDR block. • **internal**: indicates a private CIDR block. The default value is **internal**.
ip	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.

Parameter	Type	Description
ports	Array of **PortList** objects	Specifies the port mappings opened to the VPC endpoint service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share one **port_id**, either **server_port** or **protocol**, or both **server_port** and **protocol** of each of these endpoint services must be unique.
tcp_proxy	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**.
tags	Array of **TagList** objects	Specifies resource tags.
error	Array of **Error** objects	Specifies the error message returned when there is an error.
enable_policy	Boolean	Specifies whether the VPC endpoint policy can be customized. • **false**: The VPC endpoint policy cannot be customized. • **true**: The VPC endpoint policy can be customized. The default value is **false**.

Parameter	Type	Description
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**

Table 4-29 PortList

Parameter	Type	Description
client_port	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
protocol	String	Port mapping protocol. TCP is supported. Default: **TCP**

Table 4-30 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-31 Error

Parameter	Type	Description
message	String	Task error description Minimum: **0** Maximum: **1024**
code	String	Error code of a task exception. Minimum: **0** Maximum: **10**

Example Requests

Querying details about a VPC endpoint service

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

This request is to query connections of the VPC endpoint service whose ID is 4189d3c2-8882-4871-a3c2-d380272eed88.GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88
```

Example Responses

Status code: 200

The server has successfully processed the request.


```
{
  "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"
}
```


Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.4 Modifying a VPC Endpoint Service

Function

This API is used to modify a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-32 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-33 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-34 Request body parameters

Parameter	Mandatory	Type	Description
approval_enabled	No	Boolean	Specifies whether approval is required. • false: No approval is required. The created VPC endpoint is in the accepted state by default. • true: Approval is required. The created endpoint connection can be used only after being approved by the user of the VPC endpoint service. The default value is true. Default: **true**

Parameter	Mandatory	Type	Description
service_name	No	String	Specifies the name of the VPC endpoint service. The name can contain a maximum of 16 characters, including letters, digits, underscores (_), and hyphens (-). Minimum: **0** Maximum: **16**
ports	No	Array of **PortList** objects	Specifies the port mapping list of the service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same port ID, service ports and terminal ports of all these endpoint services cannot be duplicated when the protocol is the same. A maximum of 200 port mappings can be created at a time. The new value of this parameter will completely the existing one.
port_id	No	String	Specifies the ID of the NIC that ECS private IP address is bound to. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
tcp_proxy	No	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**.
description	No	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) and great than signs (>). Minimum: **0** Maximum: **512**
ip	No	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.

Table 4-35 PortList

Parameter	Mandatory	Type	Description
client_port	No	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	No	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
protocol	No	String	Port mapping protocol. TCP is supported. Default: **TCP**

Response Parameters

Status code: 200

Table 4-36 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Parameter	Type	Description
port_id	String	Specifies the ID that identifies the backend resource of the VPC endpoint service. The ID is in UUID format. The value can be: • **LB**: indicates the port ID of the private IP address of a load balancer (recommended). • **VM**: indicates the NIC ID of an ECS IP address. • **VIP**: indicates the NIC ID of the virtual server for which the virtual IP address is configured. (This value has been discarded. **LB** is recommended.) Minimum: **1** Maximum: **64**
service_name	String	Specifies the name of the VPC endpoint service. Minimum: **0** Maximum: **128**
server_type	String	Specifies the resource type. • **VM**: cloud server • **VIP**: virtual IP address • **LB**: enhanced load balancer
vpc_id	String	Specifies the ID of the VPC where the backend resource of the VPC endpoint service is located. Minimum: **1** Maximum: **64**
pool_id	String	Specifies the cluster ID of the VPC endpoint service.
approval_enabled	Boolean	Specifies whether connection approval is required. • **false**: Connection approval is not required. The created VPC endpoint is in the **accepted** state. • **true**: Connection approval is required. The created VPC endpoint is in the **pendingAcceptance** state, and it can be used only after being approved by the user of the VPC endpoint service.

Parameter	Type	Description
status	String	Specifies the status of the VPC endpoint service. • **creating**: The VPC endpoint service is being created. • **available**: The VPC endpoint service is connectable. • **failed**: The VPC endpoint service failed to be created.
service_type	String	Specifies the type of the VPC endpoint service. There are two types of VPC endpoint services: interface and gateway. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can create VPC endpoints to connect to gateway and interface VPC endpoint services. Minimum: **1** Maximum: **16**
created_at	String	Specifies when the VPC endpoint service was created. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
updated_at	String	Specifies when the VPC endpoint service was updated. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
project_id	String	Specifies the project ID. Minimum: **1** Maximum: **64**
ip	String	Specifies the IPv4 address or domain name of the interface VPC endpoint in VLAN scenarios.
ports	Array of **PortList** objects	Specifies the port mappings opened to the VPC endpoint service. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share one **port_id**, either **server_port** or **protocol**, or both **server_port** and **protocol** of each of these endpoint services must be unique.

Parameter	Type	Description
tcp_proxy	String	Specifies whether to transfer client information, such as source IP addresses, source port numbers, and marker IDs, to the server. The information can be sent to the server in the following ways: • TCP TOA: The client information is placed into the **tcp option** field and sent to the server. Note: TCP TOA is available only when the backend resource is an OBS resource. • Proxy Protocol: The client information is placed into the **tcp payload** field and sent to the server. **tcp_proxy** is available only when the server can parse the **tcp option** and **tcp payload** fields. The value can be one of the following: • **close**: Neither TCP TOA nor Proxy Protocol information is carried. • **toa_open**: TCP TOA information is carried. • **proxy_open**: Proxy Protocol information in carried. • **open**: Both TCP TOA and Proxy Protocol information are carried. The default value is **close**.
tags	Array of **TagList** objects	Specifies resource tags.
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**
enable_policy	Boolean	Specifies whether the VPC endpoint policy can be customized. • **false**: The VPC endpoint policy cannot be customized. • **true**: The VPC endpoint policy can be customized. The default value is **false**.

Parameter	Type	Description
ip_version	String	Specifies the IP version of the VPC endpoint service. Only professional VPC endpoint services support this parameter. ● **ipv4**: The IP address of the VPC endpoint service is an IPv4 address. ● **ipv6**: The IP address of the VPC endpoint service is an IPv6 address.

Table 4-37 PortList

Parameter	Type	Description
client_port	Integer	Specifies the port to be accessed by a VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: 1 to 65535. Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is associated with backend resources to provide VPC endpoint services. Supported range: 1 to 65535 Minimum: **1** Maximum: **65535** Minimum: **1** Maximum: **65535**
protocol	String	Port mapping protocol. TCP is supported. Default: **TCP**

Table 4-38 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Example Requests

Modifying a VPC endpoint service (Setting **approval_enabled** to **true**, **client_port** to **8081** and **8082**, and **server_port** to **22** and **23**)

```
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"
  } ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

Modifying a VPC endpoint service (Setting **approval_enabled** to **true**, **client_port** to **8081** and **8082**, and **server_port** to **22** and **23**)

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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 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");

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

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

Modifying a VPC endpoint service (Setting **approval_enabled** to **true**, **client_port** to **8081** and **8082**, and **server_port** to **22** and **23**)

```
# 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 = UpdateEndpointServiceRequest()
        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

Modifying a VPC endpoint service (Setting **approval_enabled** to **true**, **client_port** to **8081** and **8082**, and **server_port** to **22** and **23**)

```
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.UpdateEndpointServiceRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.5 Deleting a VPC Endpoint Service

Function

This API is used to delete a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-39 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-40 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

None

Example Requests

Deleting a VPC endpoint service

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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



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

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

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

    credentials = BasicCredentials(ak, sk)

    client = VpcepClient.new_builder() \
```



```
        .with_credentials(credentials) \
        .with_region(VpcepRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteEndpointServiceRequest()
        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")

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

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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.6 Querying Connections to a VPC Endpoint Service

Function

This API is used to query connections to a VPC endpoint service. The marker ID is the unique ID of each connection.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-41 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Table 4-42 Query Parameters

Parameter	Mandatory	Type	Description
id	No	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
marker_id	No	String	Specifies the packet ID of the VPC endpoint.
status	No	String	Specifies the connection status of the VPC endpoint. ● pendingAcceptance: The VPC endpoint connection is to be accepted. ● creating: The VPC endpoint connection is being created. ● accepted: The VPC endpoint connection has been accepted. ● failed: The VPC endpoint connection failed.
sort_key	No	String	Specifies the sorting field of the VPC endpoints, which can be: ●create_at: VPC endpoints are sorted by creation time. ●update_at: VPC endpoints are sorted by update time. The default field is create_at. Default: **create_at**
sort_dir	No	String	Specifies the sorting method of VPC endpoints, which can be: ● desc: VPC endpoints are sorted in descending order. ● asc: VPC endpoints are sorted in ascending order. The default method is desc. Default: **desc**
limit	No	Integer	Specifies the maximum number of connections displayed on each page. The value ranges from 0 to 1000 and is generally 10, 20, or 50. The default number is 10. Minimum: **1** Maximum: **1000** Default: **10**
offset	No	Integer	Specifies the offset. All VPC endpoint services after this offset will be queried. The offset must be an integer greater than 0 but less than the number of VPC endpoint services. Minimum: **0**

Request Parameters

Table 4-43 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-44 Response body parameters

Parameter	Type	Description
connections	Array of **ConnectionEndpoints** objects	Specifies the connection list.
total_count	Integer	Specifies the total number of VPC endpoints that meet the search criteria. The number is not affected by **limit** or **offset**.

Table 4-45 ConnectionEndpoints

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
marker_id	Integer	Specifies the packet ID of the VPC endpoint.

Parameter	Type	Description
created_at	String	Specifies the creation time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
domain_id	String	Domain ID of a user. Minimum: **1** Maximum: **64**
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of status is failed.
status	String	Specifies the connection status of the VPC endpoint. ● pendingAcceptance: The VPC endpoint connection is to be accepted. ● creating: The VPC endpoint connection is being created. ● accepted: The VPC endpoint connection has been accepted. ● rejected: The VPC endpoint connection has been rejected. ● failed: The VPC endpoint connection failed. ● deleting: The VPC endpoint connection is being deleted.
description	String	Specifies the description of a VPC endpoint connection. Minimum: **0** Maximum: **512**

Table 4-46 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Example Requests

Querying connections to a VPC endpoint service

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}&limit={limit}&offset={offset}

This request is to query connections of the VPC endpoint service whose ID is 4189d3c2-8882-4871-a3c2-d380272eed88. GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/connections

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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
}
```

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.7 Accepting or Rejecting a VPC Endpoint

Function

This API is used to accept or reject a VPC endpoint for a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-47 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-48 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-49 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies whether to accept or reject a VPC endpoint for a VPC endpoint service. ● receive: The connection is allowed. ● reject: The connection is rejected.

Parameter	Mandatory	Type	Description
endpoints	Yes	Array of strings	Specifies VPC endpoint IDs. Each request accepts or rejects only one VPC endpoint. Array Length: **1 - 1**

Response Parameters

Status code: 200

Table 4-50 Response body parameters

Parameter	Type	Description
connections	Array of **ConnectionEndpoints** objects	Specifies the connection list.

Table 4-51 ConnectionEndpoints

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
created_at	String	Specifies the creation time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
domain_id	String	Domain ID of a user. Minimum: **1** Maximum: **64**
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of status is failed.

Parameter	Type	Description
status	String	Specifies the connection status of the VPC endpoint. ● pendingAcceptance: The VPC endpoint connection is to be accepted. ● creating: The VPC endpoint connection is being created. ● accepted: The VPC endpoint connection has been accepted. ● rejected: The VPC endpoint connection has been rejected. ● failed: The VPC endpoint connection failed. ● deleting: The VPC endpoint connection is being deleted.
description	String	Specifies the description of a VPC endpoint connection. Minimum: **0** Maximum: **512**

Table 4-52 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Example Requests

Accepting a connection from a VPC endpoint

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

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

Accepting a connection from a VPC endpoint

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

Accepting a connection from a VPC endpoint

```
# 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 = AcceptOrRejectEndpointRequest()
        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

Accepting a connection from a VPC endpoint

```
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.AcceptOrRejectEndpointRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.8 Querying Whitelist Records of a VPC Endpoint Service

Function

This API is used to query whitelist records of a VPC endpoint service. Note Your account is in the whitelist of your own VPC endpoint service by default.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-53 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Table 4-54 Query Parameters

Parameter	Mandatory	Type	Description
permission	No	String	Specifies the permission account ID in iam:domain::domain_id format. domain_id indicates the account ID of the authorized user, for example, iam:domain::6e9dfd51d1124e8d8498dce894923a0d. Fuzzy search is supported. Minimum: **1** Maximum: **1000**
limit	No	Integer	Specifies the number of returned whitelist records of the VPC endpoint service on each page. The value ranges from 0 to 500 and is generally 10, 20, or 50. The default number is 10. Minimum: **1** Maximum: **500** Default: **10**

Parameter	Mandatory	Type	Description
offset	No	Integer	Specifies the offset. All VPC endpoint services after this offset will be queried. The offset must be an integer greater than 0 but less than the number of VPC endpoint services. Minimum: **0**
sort_key	No	String	Specifies the sorting field of whitelist records. The value is create_at, indicating the time when the whitelist record is added. Default: **create_at**
sort_dir	No	String	Specifies the sorting method of whitelist records, which can be: ● desc: The whitelist records are sorted in descending order. ● asc: The whitelist records are sorted in ascending order. The default method is desc. Default: **desc**

Request Parameters

Table 4-55 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-56 Response body parameters

Parameter	Type	Description
permissions	Array of **PermissionObject** objects	Specifies the permissions.
total_count	Integer	Specifies the total number of whitelist records that meet the search criteria. The number is not affected by **limit** or **offset**.

Table 4-57 PermissionObject

Parameter	Type	Description
id	String	Specifies the unique ID of the permission. Minimum: **1** Maximum: **64**

Parameter	Type	Description
permission	String	Specifies the permissions. The permission formats are as follows: • **iam:domain::domain_id iam:domain::** is a fixed format. *domain_id* indicates the account ID of the user that can be connected. *domain_id* can contain a maximum of 64 characters, including only letters and digits. • **organizations:orgPath::org_path organizations:orgPath::** is a fixed format. *org_path* indicates the organization path of the user that can be connected. *org_path* can contain a maximum of 1,024 characters, including only letters, digits, forward slashes (/), hyphens (-), and question marks (?). You can also enter an asterisk (*) for *domain_id* or *org_path*, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Examples: • **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** • **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **1** Maximum: **1024**
permission_type	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.
created_at	String	Specifies the time of adding the whitelist record. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.

Example Requests

Querying whitelist records of a VPC endpoint service

```
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}
```


This request is to query whitelist records of the VPC endpoint service whose ID is 4189d3c2-8882-4871-a3c2-d380272eed88. GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions

Example Responses

Status code: 200

The server has successfully processed the request.

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

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.9 Batch Adding or Deleting Whitelist Records of a VPC Endpoint Service

Function

This API is used to batch add or delete whitelist records of a VPC endpoint service. Note Your account is in the whitelist of your own VPC endpoint service by default.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-58 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-59 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-60 Request body parameters

Parameter	Mandatory	Type	Description
permissions	Yes	Array of strings	Specifies the permissions. The permission formats are as follows: • **iam:domain::domain_id** **iam:domain::** is a fixed format. *domain_id* indicates the account ID of the user that can be connected. *domain_id* can contain a maximum of 64 characters, including only letters and digits. • **organizations:orgPath::org_path** **organizations:orgPath::** is a fixed format. *org_path* indicates the organization path of the user that can be connected. *org_path* can contain a maximum of 1,024 characters, including only letters, digits, forward slashes (/), hyphens (-), and question marks (?). You can also enter an asterisk (*) for *domain_id* or *org_path*, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Examples: • **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** • **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **1** Maximum: **1024** Array Length: **1 - 50**

Parameter	Mandatory	Type	Description
permission_type	No	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.
action	Yes	String	Specifies the action to be performed, which can be add or remove. Minimum: **1** Maximum: **32**

Response Parameters

Status code: 200

Table 4-61 Response body parameters

Parameter	Type	Description
permissions	Array of strings	Specifies the permissions. The permission formats are as follows: • **iam:domain::domain_id iam:domain::** is a fixed format. *domain_id* indicates the account ID of the user that can be connected. *domain_id* can contain a maximum of 64 characters, including only letters and digits. • **organizations:orgPath::org_path organizations:orgPath::** is a fixed format. *org_path* indicates the organization path of the user that can be connected. *org_path* can contain a maximum of 1,024 characters, including only letters, digits, forward slashes (/), hyphens (-), and question marks (?). You can also enter an asterisk (*) for *domain_id* or *org_path*, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Examples: • **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** • **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **0** Maximum: **32**
permission_type	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.

Example Requests

- Adding a Whitelist for a VPC Endpoint Service

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

{
  "permissions" : [ "iam:domain::fc973eea581490997e82ea11a1d0101" ],
```



```
 "action" : "add"
}
```

- Removing a VPC Endpoint Service from the Whitelist

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

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

Example Responses

Status code: 200

The server has successfully processed the request.

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

SDK Sample Code

The SDK sample code is as follows.

Java

- Adding a Whitelist for a VPC Endpoint Service

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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



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

- Removing a VPC Endpoint Service from the Whitelist

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

- Adding a Whitelist for a VPC Endpoint Service

```
# 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 = AddOrRemoveServicePermissionsRequest()
        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)
```

- Removing a VPC Endpoint Service from the Whitelist

```
# 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 = AddOrRemoveServicePermissionsRequest()
        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

- Adding a Whitelist for a VPC Endpoint Service

```
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.AddOrRemoveServicePermissionsRequest{}
    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)
    }
}
```

- Removing a VPC Endpoint Service from the Whitelist

```
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.AddOrRemoveServicePermissionsRequest{}
    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)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.10 Querying Public VPC Endpoint Services

Function

This API is used to query public VPC endpoint services. These services are created by O&M personnel and can be visible to and assessed by all users.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-62 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Table 4-63 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the maximum number of public VPC endpoint services displayed on each page. The value ranges from 0 to 1000 and is generally 10, 20, or 50. The default number is 10. Minimum: **1** Maximum: **1000** Default: **10**
offset	No	Integer	Specifies the offset. All VPC endpoint services after this offset will be queried. The offset must be an integer greater than 0 but less than the number of VPC endpoint services. Minimum: **0**
endpoint_service_name	No	String	Specifies the name of the public VPC endpoint service. The value is case-sensitive and supports fuzzy match. Minimum: **1** Maximum: **128**
id	No	String	Unique ID of the public VPC endpoint service. Minimum: **1** Maximum: **64**
sort_key	No	String	Specifies the sorting field of the VPC endpoint services, which can be: ● create_at: VPC endpoint services are sorted by creation time. ● update_at: VPC endpoint services are sorted by update time. The default field is create_at. Default: **create_at**

Parameter	Mandatory	Type	Description
sort_dir	No	String	Specifies the sorting method of VPC endpoint services, which can be: ● desc: VPC endpoint services are sorted in descending order. ● asc: VPC endpoint services are sorted in ascending order. The default method is desc. Default: **desc**

Request Parameters

Table 4-64 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-65 Response body parameters

Parameter	Type	Description
endpoint_services	Array of **EndpointService** objects	Specifies VPC endpoint services.
total_count	Integer	Specifies the total number of public VPC endpoint services that meet the search criteria. The number is not affected by **limit** or **offset**.

Table 4-66 EndpointService

Parameter	Type	Description
id	String	Unique ID of the public VPC endpoint service. Minimum: **1** Maximum: **64**
owner	String	Specifies the owner of the VPC endpoint service.
service_name	String	Specifies the name of the public VPC endpoint service.
service_type	String	Specifies the type of the VPC endpoint service. ● gateway: VPC endpoint services of this type are configured by O&M personnel. You can use them directly without creating them by yourselves. ● interface: Cloud services configured by O&M personnel and private services created by yourselves are included. You cannot configure these cloud services, but can use them. You can create a VPC endpoint for accessing gateway and interface VPC endpoint services.
created_at	String	Specifies the creation time of the VPC endpoint service. The applied UTC time format is YYYYMM-DDTHH:MM:SSZ.
is_charge	Boolean	Specifies whether the associated VPC endpoint carries a charge. ● true: The associated VPC endpoint needs to be charged. ● false: The associated VPC endpoint does not need to be charged.
enable_policy	Boolean	Specifies whether the VPC endpoint policy is enabled. ● **false**: The VPC endpoint policy cannot be set. ● **true**: The VPC endpoint policy can be set. The default value is **false**. Specifies whether the VPC endpoint policy is enabled. ● **false**: The VPC endpoint policy cannot be set. ● **true**: The VPC endpoint policy can be set. The default value is **false**.

Example Requests

Querying public VPC endpoint services

```
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}
```


Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        ListServicePublicDetailsRequest request = new ListServicePublicDetailsRequest();
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withEndpointServiceName("<endpoint_service_name>");
        request.withId("<id>");
```



```
        request.withSortKey(ListServicePublicDetailsRequest.SortKeyEnum.fromValue("<sort_key>"));
        request.withSortDir(ListServicePublicDetailsRequest.SortDirEnum.fromValue("<sort_dir>"));
        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.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 = ListServicePublicDetailsRequest()
        request.limit = <limit>
        request.offset = <offset>
        request.endpoint_service_name = "<endpoint_service_name>"
        request.id = "<id>"
        request.sort_key = "<sort_key>"
        request.sort_dir = "<sort_dir>"
        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")

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

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

    request := &model.ListServicePublicDetailsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    endpointServiceNameRequest:= "<endpoint_service_name>"
    request.EndpointServiceName = &endpointServiceNameRequest
    idRequest:= "<id>"
    request.Id = &idRequest
    sortKeyRequest:= model.GetListServicePublicDetailsRequestSortKeyEnum().<SORT_KEY>
    request.SortKey = &sortKeyRequest
    sortDirRequest:= model.GetListServicePublicDetailsRequestSortDirEnum().<SORT_DIR>
    request.SortDir = &sortDirRequest
    response, err := client.ListServicePublicDetails(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.11 Querying Basic Information About a VPC Endpoint Service

Function

This API is used to query basic information about a VPC endpoint service. You can use this API to query the target VPC endpoint service. This API can also be used by other users to query basic information about your VPC endpoint service, without exposing your server information.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-67 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Table 4-68 Query Parameters

Parameter	Mandatory	Type	Description
endpoint_service_name	No	String	Specifies the name of the VPC endpoint service. Note: Either this field or the id field must be specified. Otherwise, an error occurs. Minimum: **1** Maximum: **128**
id	No	String	Specifies the ID of the VPC endpoint service. The ID is the unique identifier of the VPC endpoint service. Note: Either this parameter or endpoint_service_name must be specified. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-69 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-70 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
service_name	String	Specifies the name of the VPC endpoint service.

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service. Only your private services can be configured into interface VPC endpoint services. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can create VPC endpoints to connect to gateway and interface VPC endpoint services.
created_at	String	Specifies when the VPC endpoint service was created. The UTC time format *YYYY-MM-DDTHH:MM:SSZ* is used.
is_charge	Boolean	Specifies whether the VPC endpoint connected to the VPC endpoint service is billed. • **true**: The VPC endpoint is billed. • **false**: The VPC endpoint is not billed.
public_border_group	String	Specifies the public border group information about the pool corresponding to the VPC endpoint.
enable_policy	Boolean	Specifies whether the VPC endpoint policy can be customized. • **false**: The VPC endpoint policy cannot be customized. • **true**: The VPC endpoint policy can be customized. The default value is **false**.

Example Requests

Querying basic information about a VPC endpoint service

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "id" : "9d4c1028-1336-4556-9881-b5d807c1b8a8",
  "service_name" : "test123",
```



```
  "service_type" : "interface",
  "created_at" : "2018-09-17T07:28:31Z",
  "is_charge" : "true"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        ListServiceDescribeDetailsRequest request = new ListServiceDescribeDetailsRequest();
        request.withEndpointServiceName("<endpoint_service_name>");
        request.withId("<id>");
        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"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListServiceDescribeDetailsRequest()
        request.endpoint_service_name = "<endpoint_service_name>"
        request.id = "<id>"
        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")

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

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

    request := &model.ListServiceDescribeDetailsRequest{}
    endpointServiceNameRequest:= "<endpoint_service_name>"
    request.EndpointServiceName = &endpointServiceNameRequest
    idRequest:= "<id>"
```



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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.12 Changing the Name of a VPC Endpoint Service

Function

This API is used to change the name of a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-71 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the tenant ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-72 Request body parameters

Parameter	Mandatory	Type	Description
endpoint_service_name	No	String	Specifies the name of the VPC endpoint service.

Response Parameters

Status code: 200

Table 4-73 Response body parameters

Parameter	Type	Description
endpoint_service_name	String	Specifies the name of the VPC endpoint service.

Example Requests

Changing the name of a VPC endpoint service

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

{
  "endpoint_service_name" : "obs_test"
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Changing the name of a VPC endpoint service

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

Changing the name of a VPC endpoint service

```
# 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 = UpdateEndpointServiceNameRequest()
        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

Changing the name of a VPC endpoint service

```
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.UpdateEndpointServiceNameRequest{}
    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)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.13 Updating Descriptions of VPC Endpoint Connections

Function

This API is used to update descriptions of VPC endpoint connections.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-74 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the tenant ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-75 Request body parameters

Parameter	Mandatory	Type	Description
connections	Yes	Array of **Connections Desc** objects	Specifies the VPC endpoint connections.

Table 4-76 ConnectionsDesc

Parameter	Mandatory	Type	Description
id	Yes	String	Specifies the VPC endpoint ID in UUID format. Minimum: **1** Maximum: **64**
description	Yes	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) and great than signs (>). Minimum: **0** Maximum: **512**

Response Parameters

Status code: 200

Table 4-77 Response body parameters

Parameter	Type	Description
connections	Array of **ConnectionEndpoints** objects	Specifies the connection list.

Table 4-78 ConnectionEndpoints

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
created_at	String	Specifies the creation time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The applied UTC time format is YYYY-MMDDTHH:MM:SSZ.
domain_id	String	Domain ID of a user. Minimum: **1** Maximum: **64**
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of status is failed.
status	String	Specifies the connection status of the VPC endpoint. ● pendingAcceptance: The VPC endpoint connection is to be accepted. ● creating: The VPC endpoint connection is being created. ● accepted: The VPC endpoint connection has been accepted. ● rejected: The VPC endpoint connection has been rejected. ● failed: The VPC endpoint connection failed. ● deleting: The VPC endpoint connection is being deleted.
description	String	Specifies the description of a VPC endpoint connection. Minimum: **0** Maximum: **512**

Table 4-79 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**

Parameter	Type	Description
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Example Requests

Updating the description of a VPC endpoint connection

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

{
  "connections" : [ {
    "id" : "0df78897-35b0-4a93-ad17-0bc0585f546b",
    "description" : "Specifies the VPC endpoint 1, which belongs to account
[0605767a3300d5762fb7c0186d9e1779]abc_26."
  }, {
    "id" : "0d837f75-90d5-4528-bd11-5d974238ebca",
    "description" : "Specifies the VPC endpoint 2, which belongs to account [7876999999999]abc_26."
  } ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "connections" : [ {
    "id" : "0d837f75-90d5-4528-bd11-5d974238ebca",
    "status" : "accepted",
    "description" : "Specifies the VPC endpoint 2, which belongs to account [7876999999999]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" : "Specifies the VPC endpoint 1, which belongs to account
[0605767a3300d5762fb7c0186d9e1779]abc_26.",
    "marker_id" : 302001833,
    "domain_id" : "05b5408a0a80d2b10f06c0184a774460",
    "created_at" : "2022-04-14T01:59:31Z",
    "updated_at" : "2022-04-14T07:57:39Z"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the description of a VPC endpoint connection

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



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

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateEndpointConnectionsDescRequest request = new UpdateEndpointConnectionsDescRequest();
        UpdateEndpointConnectionsDescRequestBody body = new UpdateEndpointConnectionsDescRequestBody();
        List<ConnectionsDesc> listbodyConnections = new ArrayList<>();
        listbodyConnections.add(
            new ConnectionsDesc()
                .withId("0df78897-35b0-4a93-ad17-0bc0585f546b")
                .withDescription("Specifies the VPC endpoint 1, which belongs to account [0605767a3300d5762fb7c0186d9e1779]abc_26.")
        );
        listbodyConnections.add(
            new ConnectionsDesc()
                .withId("0d837f75-90d5-4528-bd11-5d974238ebca")
                .withDescription("Specifies the VPC endpoint 2, which belongs to account [7876999999999]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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

Updating the description of a VPC endpoint connection

```
# 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 = UpdateEndpointConnectionsDescRequest()
        listConnectionsbody = [
            ConnectionsDesc(
                id="0df78897-35b0-4a93-ad17-0bc0585f546b",
                description="Specifies the VPC endpoint 1, which belongs to account [0605767a3300d5762fb7c0186d9e1779]abc_26."
            ),
            ConnectionsDesc(
                id="0d837f75-90d5-4528-bd11-5d974238ebca",
                description="Specifies the VPC endpoint 2, which belongs to account [7876999999999]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

Updating the description of a VPC endpoint connection

```
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.UpdateEndpointConnectionsDescRequest{}
    var listConnectionsbody = []model.ConnectionsDesc{
        {
            Id: "0df78897-35b0-4a93-ad17-0bc0585f546b",
            Description: "Specifies the VPC endpoint 1, which belongs to account [0605767a3300d5762fb7c0186d9e1779]abc_26.",
        },
        {
            Id: "0d837f75-90d5-4528-bd11-5d974238ebca",
            Description: "Specifies the VPC endpoint 2, which belongs to account [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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.14 Batch Adding Whitelist Records of a VPC Endpoint Service

Function

This API is used to batch add whitelist records for VPC endpoint services of the current user. The description can be added. Note Your account is in the whitelist of your own VPC endpoint service by default.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-80 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-81 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.

Parameter	Mandatory	Type	Description
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-82 Request body parameters

Parameter	Mandatory	Type	Description
permissions	Yes	Array of **EpsAddPermissionRequest** objects	Specifies the whitelist records of the VPC endpoint service.
permission_type	No	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.

Table 4-83 EpsAddPermissionRequest

Parameter	Mandatory	Type	Description
permission	Yes	String	The permission format is **iam:domain::domain_id** or **organizations:orgPath::org_path**. • **iam:domain::** and **organizations:orgPath::** are fixed formats. • *domain_id* indicates the ID of the account in which VPC endpoints can be created to access the VPC endpoint service. *org_path* indicates the organization path under which accounts can create VPC endpoints to access the VPC endpoint service. *domain_id* can contain a maximum of 64 characters, including only letters and digits. *org_path* can contain a maximum of 1,024 characters, including only letters, digits, forward slashes (/), and hyphens (-). Example: **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** or **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **0** Maximum: **1024**
description	Yes	String	Specifies the description of a whitelist record of a VPC endpoint service. Minimum: **0** Maximum: **256**

Response Parameters

Status code: 200

Table 4-84 Response body parameters

Parameter	Type	Description
permissions	Array of **EpsPermission** objects	Specifies the whitelist of the VPC endpoint service.

Table 4-85 EpsPermission

Parameter	Type	Description
id	String	Specifies primary key IDs of whitelist records of a VPC endpoint service.
permission	String	The permission format is **iam:domain::domain_id** or **organizations:orgPath::org_path**. • **iam:domain::** and **organizations:orgPath::** are fixed formats. • *domain_id* indicates the ID of the account in which VPC endpoints can be created to access the VPC endpoint service. *org_path* indicates the organization path under which accounts can create VPC endpoints to access the VPC endpoint service. *domain_id* can contain only letters and digits. *org_path* can contain only letters, digits, forward slashes (/), and hyphens (-). You can also enter an asterisk (*) for domain_id or org_path, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Example: **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** or **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **1** Maximum: **1024**

Parameter	Type	Description
permission_type	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.
description	String	Specifies the description of a whitelist record of a VPC endpoint service. Minimum: **0** Maximum: **256**
created_at	String	Specifies the time when a whitelist record is created.

Example Requests

Batch adding whitelist records of a VPC endpoint service

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

{
  "permissions" : [ {
    "permission" : "*",
    "description" : "Specifies APIs for batch adding whitelist records of a VPC endpoint service."
  }, {
    "permission" : "iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",
    "description" : "test"
  } ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "permissions" : [ {
    "id" : "ab42a58b-6f8e-43b3-9400-aaf0e393ee0d",
    "permission" : "iam:domain::4dbb0ce8766f44a9bfd08ccf8fc02397",
    "description" : "test",
    "created_at" : "2022-06-16T10:03:27Z"
  }, {
    "id" : "dfaaeff1-858d-4631-83c4-be3548a07935",
    "permission" : "*",
    "description" : "",
    "created_at" : "2022-06-16T10:03:27Z"
  } ]
}
```


SDK Sample Code

The SDK sample code is as follows.

Java

Batch adding whitelist records of a VPC endpoint service

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchAddEndpointServicePermissionsRequest request = new BatchAddEndpointServicePermissionsRequest();
        BatchAddEndpointServicePermissionsRequestBody body = new BatchAddEndpointServicePermissionsRequestBody();
        List<EpsAddPermissionRequest> listbodyPermissions = new ArrayList<>();
        listbodyPermissions.add(
            new EpsAddPermissionRequest()
                .withPermission("*")
                .withDescription("Specifies APIs for batch adding whitelist records of a VPC endpoint service.")
        );
        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

Batch adding whitelist records of a VPC endpoint service

```
# 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 = BatchAddEndpointServicePermissionsRequest()
        listPermissionsbody = [
            EpsAddPermissionRequest(
                permission="*",
                description="Specifies APIs for batch adding whitelist records of a VPC endpoint service."
            ),
            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

Batch adding whitelist records of a VPC endpoint service

```
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.BatchAddEndpointServicePermissionsRequest{}
    var listPermissionsbody = []model.EpsAddPermissionRequest{
        {
            Permission: "*",
            Description: "Specifies APIs for batch adding whitelist records of a VPC endpoint service.",
        },
        {
            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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.15 Batch Deleting Whitelist Records of a VPC Endpoint Service

Function

This API is used to batch delete whitelist records of a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-86 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-87 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-88 Request body parameters

Parameter	Mandatory	Type	Description
permissions	Yes	Array of **EpsRemovePermissionRequest** objects	Specifies the whitelist of the VPC endpoint service.

Table 4-89 EpsRemovePermissionRequest

Parameter	Mandatory	Type	Description
id	Yes	String	Specifies primary key IDs of whitelist records of a VPC endpoint service. Minimum: **1** Maximum: **64**

Response Parameters

Status code: 200

Table 4-90 Response body parameters

Parameter	Type	Description
permissions	Array of **EpsPermission** objects	Specifies the whitelist of the VPC endpoint service.

Table 4-91 EpsPermission

Parameter	Type	Description
id	String	Specifies primary key IDs of whitelist records of a VPC endpoint service.

Parameter	Type	Description
permission	String	The permission format is **iam:domain::domain_id** or **organizations:orgPath::org_path**. • **iam:domain::** and **organizations:orgPath::** are fixed formats. • *domain_id* indicates the ID of the account in which VPC endpoints can be created to access the VPC endpoint service. *org_path* indicates the organization path under which accounts can create VPC endpoints to access the VPC endpoint service. *domain_id* can contain only letters and digits. *org_path* can contain only letters, digits, forward slashes (/), and hyphens (-). You can also enter an asterisk (*) for domain_id or org_path, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Example: **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd** or **organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **1** Maximum: **1024**
permission_type	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.
description	String	Specifies the description of a whitelist record of a VPC endpoint service. Minimum: **0** Maximum: **256**
created_at	String	Specifies the time when a whitelist record is created.

Example Requests

Batch deleting whitelist records of a VPC endpoint service


```
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"
  } ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

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

SDK Sample Code

The SDK sample code is as follows.

Java

Batch deleting whitelist records of a VPC endpoint service

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

Batch deleting whitelist records of a VPC endpoint service

```
# 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 = BatchRemoveEndpointServicePermissionsRequest()
        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

Batch deleting whitelist records of a VPC endpoint service

```
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.BatchRemoveEndpointServicePermissionsRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.2.16 Updating the Description of a Whitelist Record of a VPC Endpoint Service

Function

This API is used to update the whitelist description of a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-92 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
permission_id	Yes	String	Specifies IDs of whitelist records of a VPC endpoint service. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-93 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-94 Request body parameters

Parameter	Mandatory	Type	Description
permission	Yes	**EpsUpdatePermissionDesc** object	Specifies the request body for updating the description of a whitelist record of a VPC endpoint service.

Table 4-95 EpsUpdatePermissionDesc

Parameter	Mandatory	Type	Description
description	Yes	String	Specifies the description of a whitelist record of a VPC endpoint service. Minimum: **0** Maximum: **256**

Response Parameters

Status code: 200

Table 4-96 Response body parameters

Parameter	Type	Description
permissions	Array of **EpsPermission** objects	Specifies the whitelist of the VPC endpoint service.

Table 4-97 EpsPermission

Parameter	Type	Description
id	String	Specifies primary key IDs of whitelist records of a VPC endpoint service.
permission	String	The permission format is **iam:domain::domain_id** or **organizations:orgPath::org_path**. • **iam:domain::** and **organizations:orgPath::** are fixed formats. • *domain_id* indicates the ID of the account in which VPC endpoints can be created to access the VPC endpoint service. *org_path* indicates the organization path under which accounts can create VPC endpoints to access the VPC endpoint service. *domain_id* can contain only letters and digits. *org_path* can contain only letters, digits, forward slashes (/), and hyphens (-). You can also enter an asterisk (*) for domain_id or org_path, which indicates that this VPC endpoint service allows accesses from any VPC endpoint. Example: ***iam:domain::6e9dfd51d1124e8d8498dce894923a0dd*** *or* ***organizations:orgPath::o-3j59d1231uprgk9yuvlidra7zbzfi578/r-rldbu1vmxdw5ahdkknxnvd5rgag77m2z/ou-7tuddd8nh99rebxltawsm6qct5z7rklv/*** Minimum: **1** Maximum: **1024**

Parameter	Type	Description
permission_type	String	Specifies the whitelist type of the VPC endpoint service. • **domainId**: indicates the whitelisted ID of the account that can create VPC endpoints to connect to the VPC endpoint service. • **orgPath**: indicates the whitelisted organization path under which accounts can create VPC endpoints to connect to the VPC endpoint service.
description	String	Specifies the description of a whitelist record of a VPC endpoint service. Minimum: **0** Maximum: **256**
created_at	String	Specifies the time when a whitelist record is created.

Example Requests

Updating the description of a whitelist record of a VPC endpoint service

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

{
  "permission" : {
    "description" : "Specifies the whitelist record to the VPC endpoint service added."
  }
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "permissions" : [ {
    "id" : "849046d9-7342-44ba-b2d2-367d8de6311b",
    "permission" : "*",
    "description" : "Specifies the whitelist record to the VPC endpoint service added.",
    "created_at" : "2022-06-16T09:48:21Z"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the description of a whitelist record of a VPC endpoint service


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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateEndpointServicePermissionDescRequest request = new UpdateEndpointServicePermissionDescRequest();
        UpdateEndpointServicePermissionDescRequestBody body = new UpdateEndpointServicePermissionDescRequestBody();
        EpsUpdatePermissionDesc permissionbody = new EpsUpdatePermissionDesc();
        permissionbody.withDescription("Specifies the whitelist record to the VPC endpoint service added.");
        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

Updating the description of a whitelist record of a VPC endpoint service

```
# 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 = UpdateEndpointServicePermissionDescRequest()
        permissionbody = EpsUpdatePermissionDesc(
            description="Specifies the whitelist record to the VPC endpoint service added."
        )
        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

Updating the description of a whitelist record of a VPC endpoint service

```
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.UpdateEndpointServicePermissionDescRequest{}
    permissionbody := &model.EpsUpdatePermissionDesc{
        Description: "Specifies the whitelist record to the VPC endpoint service added.",
```



```
    }
    request.Body = &model.UpdateEndpointServicePermissionDescRequestBody{
        Permission: permissionbody,
    }
    response, err := client.UpdateEndpointServicePermissionDesc(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3 VPC Endpoints

4.3.1 Creating a VPC Endpoint

Function

This API is used to create a VPC endpoint for accessing a VPC endpoint service.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/vpc-endpoints

Table 4-98 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-99 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-100 Request body parameters

Parameter	Mandatory	Type	Description
subnet_id	No	String	Note: This parameter is mandatory to create an interface VPC endpoint. The ID must be the ID of the subnet created in the VPC specified by vpc_id and in the format of UUID. For details, see the id field in the response information of the section "Querying Subnet Details" in the Virtual Private Cloud API Reference. This parameter is mandatory when you create a VPC endpoint for connecting an interface VPC endpoint service. Note: ● The subnet CIDR block of the VPC cannot overlap with 198.19.128.0/17. ● The destination address of the custom route in the VPC route table cannot overlap with 198.19.128.0/17. Minimum: **1** Maximum: **64**
endpoint_service_id	Yes	String	Specifies the ID of the VPC endpoint service. You can obtain the ID of the VPC endpoint service to be connected by querying the section "VPCEP Service Overview". Minimum: **1** Maximum: **64**
vpc_id	Yes	String	Specifies the ID of the VPC where the VPC endpoint is located. For details, see the id field in the response information of the section "Querying VPC Details" in the Elastic Cloud Server API Reference. Minimum: **1** Maximum: **64**

Parameter	Mandatory	Type	Description
enable_dns	No	Boolean	Specifies whether to create a domain name. ● true: A domain name is required. ● false: A domain name is not required. The default value is false. Note: When you create a VPC endpoint for connecting a gateway VPC endpoint service, no domain name is created if enable_dns is set to true or false. Default: **false**
tags	No	Array of **TagList** objects	Specifies the tag list. If there is no tag in the list, tags is taken as an empty array.
routetables	No	Array of strings	Specifies the IDs of route tables. For details, see the id field in the response information of the section "Querying a VPC Route" in the Virtual Private Cloud API Reference. This parameter is mandatory when you create a VPC endpoint for connecting a gateway VPC endpoint service. Note: If this parameter is not configured, use the default route table. Minimum: **0** Maximum: **64** Array Length: **1 - 10**
port_ip	No	String	Specifies the IP address for accessing the associated VPC endpoint service. You can specify IP addresses for accessing the associated VPC endpoint service when creating a VPC endpoint. Only IPv4 addresses are supported. This parameter is mandatory when you create a VPC endpoint for connecting an interface VPC endpoint service. Minimum: **1** Maximum: **128**

Parameter	Mandatory	Type	Description
whitelist	No	Array of strings	Specifies whitelists added for controlling access to the VPC endpoint. IPv4 addresses or CIDR blocks can be specified to control access when you create a VPC endpoint. The whitelist is left blank by default. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	No	Boolean	Specifies whether access control is enabled.
description	No	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) and great than signs (>). Minimum: **0** Maximum: **512**
policy_statement	No	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for Object Storage Service (OBS) and Scalable File Service (SFS) is set to **true**. Array Length: **0 - 10**
policy_document	No	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**

Parameter	Mandatory	Type	Description
ip_version	No	String	Specifies the IP version of the VPC endpoint. Only professional VPC endpoints support this parameter. • **ipv4**: The VPC endpoint IP address can only be an IPv4 address. • **dualstack**: The VPC endpoint IP address can be an IPv4 or IPv6 address.
ipv6_address	No	String	Specifies the IPv6 address of the VPC endpoint service to be connected to. When creating a VPC endpoint, you can specify the IP address for the VPC endpoint service you will connect to. If you do not specify the IP address, the IP address generated by the system will be used. Only professional VPC endpoints support this parameter.

Table 4-101 TagList

Parameter	Mandatory	Type	Description
key	No	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**

Parameter	Mandatory	Type	Description
value	No	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-102 PolicyStatement

Parameter	Mandatory	Type	Description
Effect	Yes	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Yes	Array of strings	Specifies OBS access permissions.
Resource	Yes	Array of strings	Specifies the OBS object.

Response Parameters

Status code: 200

Table 4-103 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates VPC endpoint services that are configured by the O&M personnel. You can directly use them. • **interface**: indicates cloud services configured by the O&M personnel and private services created by yourselves. You can directly use interface VPC endpoint services configured by the O&M personnel. You can query the public VPC endpoint services to view the VPC endpoint services that are configured by the O&M personnel and visible and accessible to all users. You can create interface VPC endpoint services.
status	String	Specifies the status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint failed to be created. • **deleting**: The VPC endpoint is being deleted.
ip	String	Specifies the IP address of the VPC endpoint. Minimum: **1** Maximum: **64**
active_status	Array of strings	Specifies the account status. • **frozen**: The account is frozen. • **active**: The account is unfrozen.
endpoint_service_name	String	Specifies the name of the VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Parameter	Type	Description
enable_dns	Boolean	Specifies whether to create a domain name. • **true**: Create a domain name. • **false**: Do not create a domain name. Note When you create a VPC endpoint for connecting to a gateway VPC endpoint service, no domain name is created regardless of whether **enable_dns** is set to **true** or **false**.
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in UUID format. Minimum: **1** Maximum: **64**
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**

Parameter	Type	Description
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, **false** will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Specifies the route table ID list. If this parameter is not specified, the ID of the default route table of the VPC is returned. This parameter is available when you create a VPC endpoint for connecting to a gateway VPC endpoint service. Minimum: **0** Maximum: **64**
specification_name	String	Specifies the name of the specifications.
description	String	Provides supplementary information about the VPC endpoint. The description can contain letters and digits. Greater than symbols (<) and less than symbols (>) are not allowed. Minimum: **0** Maximum: **128**
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**
enable_status	String	Specifies whether a VPC endpoint is available. • **enable**: The VPC endpoint is available. • **disable**: The VPC endpoint is not available.
endpoint_pool_id	String	(To be discarded) Specifies the ID of the cluster associated with the VPC endpoint. Minimum: **1** Maximum: **64**
public_border_group	String	Specifies the public border group information about the pool corresponding to the VPC endpoint.

Parameter	Type	Description
ipv6_address	String	Specifies the IPv6 address of the VPC endpoint. Only professional VPC endpoints support this parameter.

Table 4-104 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-105 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

Create a VPC endpoint without creating a domain name.


```
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"
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

Create a VPC endpoint without creating a domain name.

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

        ICredential auth = new BasicCredentials()
                .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.withVpcId("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

Create a VPC endpoint without creating a domain name.

```
# 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 = CreateEndpointRequest()
        request.body = CreateEndpointRequestBody(
            enable_dns=False,
            vpc_id="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

Create a VPC endpoint without creating a domain name.

```
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.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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.2 Querying VPC Endpoints

Function

This API is used to query VPC endpoints.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/vpc-endpoints

Table 4-106 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Table 4-107 Query Parameters

Parameter	Mandatory	Type	Description
endpoint_service_name	No	String	Specifies the name of the VPC endpoint service. The name is case insensitive and supports fuzzy match. Minimum: **1** Maximum: **128**

Parameter	Mandatory	Type	Description
vpc_id	No	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**
id	No	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
limit	No	Integer	Specifies the maximum number of VPC endpoints displayed on each page. The value ranges from 0 to 500 and is generally 10, 20, or 50. The default number is 10. Minimum: **1** Maximum: **1000** Default: **10**
offset	No	Integer	Specifies the offset. All VPC endpoint services after this offset will be queried. The offset must be an integer greater than 0 but less than the number of VPC endpoint services. Minimum: **0**
sort_key	No	String	Specifies the sorting field of the VPC endpoints, which can be: ●create_at: VPC endpoints are sorted by creation time. ●update_at: VPC endpoints are sorted by update time. The default field is create_at. Default: **create_at**
sort_dir	No	String	Specifies the sorting method of VPC endpoints, which can be: ● desc: VPC endpoints are sorted in descending order. ● asc: VPC endpoints are sorted in ascending order. The default method is desc. Default: **desc**

Request Parameters

Table 4-108 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-109 Response body parameters

Parameter	Type	Description
endpoints	Array of **EndpointResponseBody** objects	Specifies the VPC endpoints.
total_count	Integer	Specifies the total number of VPC endpoints that meet the search criteria. The number is not affected by **limit** or **offset**.

Table 4-110 EndpointResponseBody

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create interface VPC endpoint services.
status	String	Specifies the status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint failed to be created. • **deleting**: The VPC endpoint is being deleted.
active_status	Array of strings	Specifies the account status. • **frozen**: The account is frozen. • **active**: The account is unfrozen.
endpoint_service_name	String	Specifies the name of the VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**

Parameter	Type	Description
enable_dns	Boolean	Specifies whether to create a domain name. • **true**: Create a domain name. • **false**: Do not create a domain name. Note When you create a VPC endpoint for connecting to a gateway VPC endpoint service, no domain name is created regardless of whether **enable_dns** is set to **true** or **false**.
dns_names	Array of strings	Specifies the domain name for accessing the associated VPC endpoint service. This parameter is available when **enable_dns** is set to **true**. Minimum: **0** Maximum: **32**
ip	String	Specifies the IP address for accessing the associated VPC endpoint service. This parameter is returned only under the following conditions: • You query a VPC endpoint for accessing an interface VPC endpoint service. • Connection approval is enabled for the VPC endpoint service, and the connection has been approved. **status** of the VPC endpoint can be **accepted** or **rejected**. The **rejected** status only appears when the VPC endpoint is accepted and then rejected. Minimum: **1** Maximum: **64**
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in UUID format. Minimum: **1** Maximum: **64**
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.

Parameter	Type	Description
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of **status** is **failed**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, access control is disabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Specifies the route table ID list. If this parameter is not specified, the ID of the default route table of the VPC is returned. This parameter is available when you create a VPC endpoint for connecting to a gateway VPC endpoint service. Minimum: **0** Maximum: **64**
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**

Parameter	Type	Description
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**
endpoint_pool_id	String	(To be discarded) Specifies the ID of the cluster associated with the VPC endpoint. Minimum: **1** Maximum: **64**
public_border_group	String	Specifies the information about the public border group associated with the VPC endpoint. This parameter is returned only when the VPC endpoint is associated with an edge pool.

Table 4-111 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-112 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Table 4-113 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

Querying VPC endpoints

```
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}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

```
        request.withEndpointServiceName("<endpoint_service_name>");
        request.withVpcId("<vpc_id>");
        request.withId("<id>");
        request.withLimit(<limit>);
        request.withOffset(<offset>);
        request.withSortKey(ListEndpointsRequest.SortKeyEnum.fromValue("<sort_key>"));
        request.withSortDir(ListEndpointsRequest.SortDirEnum.fromValue("<sort_dir>"));
        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.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 = ListEndpointsRequest()
        request.endpoint_service_name = "<endpoint_service_name>"
        request.vpc_id = "<vpc_id>"
        request.id = "<id>"
        request.limit = <limit>
        request.offset = <offset>
        request.sort_key = "<sort_key>"
        request.sort_dir = "<sort_dir>"
        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")

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

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

    request := &model.ListEndpointsRequest{}
    endpointServiceNameRequest:= "<endpoint_service_name>"
    request.EndpointServiceName = &endpointServiceNameRequest
    vpcIdRequest:= "<vpc_id>"
    request.VpcId = &vpcIdRequest
    idRequest:= "<id>"
    request.Id = &idRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    sortKeyRequest:= model.GetListEndpointsRequestSortKeyEnum().<SORT_KEY>
    request.SortKey = &sortKeyRequest
    sortDirRequest:= model.GetListEndpointsRequestSortDirEnum().<SORT_DIR>
    request.SortDir = &sortDirRequest
    response, err := client.ListEndpoints(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.3 Querying Details of a VPC Endpoint

Function

This API is used to query details of a VPC endpoint.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-114 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-115 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-116 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create interface VPC endpoint services.

Parameter	Type	Description
status	String	Specifies the status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint failed to be created. • **deleting**: The VPC endpoint is being deleted.
active_status	Array of strings	Specifies the account status. • **frozen**: The account is frozen. • **active**: The account is unfrozen.
enable_status	String	Specifies whether a VPC endpoint is available. • **enable**: The VPC endpoint is available. • **disable**: The VPC endpoint is not available.
specification_name	String	Specifies the specification name of the VPC endpoint service.
endpoint_service_name	String	Specifies the name of the VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
enable_dns	Boolean	Specifies whether to create a domain name. • **true**: Create a domain name. • **false**: Do not create a domain name. Note When you create a VPC endpoint for connecting to a gateway VPC endpoint service, no domain name is created regardless of whether **enable_dns** is set to **true** or **false**.

Parameter	Type	Description
dns_names	Array of strings	Specifies the domain name for accessing the associated VPC endpoint service. This parameter is available when **enable_dns** is set to **true**. Minimum: **0** Maximum: **32**
ip	String	Specifies the IP address for accessing the associated VPC endpoint service. This parameter is returned only under the following conditions: • You query a VPC endpoint for accessing an interface VPC endpoint service. • Connection approval is enabled for the VPC endpoint service, and the connection has been approved. **status** of the VPC endpoint can be **accepted** or **rejected**. The **rejected** status only appears when the VPC endpoint is accepted and then rejected. Minimum: **1** Maximum: **64**
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in UUID format. Minimum: **1** Maximum: **64**
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.

Parameter	Type	Description
error	**QueryError** object	Exception information returned during resource query.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, access control is disabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Specifies the route table ID list. If this parameter is not specified, the ID of the default route table of the VPC is returned. This parameter is available when you create a VPC endpoint for connecting to a gateway VPC endpoint service. Minimum: **0** Maximum: **64**
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**

Parameter	Type	Description
endpoint_pool_id	String	(To be discarded) Specifies the ID of the cluster associated with the VPC endpoint. Minimum: **1** Maximum: **64**
public_border_group	String	Specifies the public border group information about the pool corresponding to the VPC endpoint.
ipv6_address	String	Specifies the IPv6 address of the VPC endpoint service to be connected to. When creating a VPC endpoint, you can specify the IP address for the VPC endpoint service you will connect to. If you do not specify the IP address, the IP address generated by the system will be used. Only professional VPC endpoints support this parameter.

Table 4-117 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-118 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Table 4-119 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

Querying details about a VPC endpoint

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListEndpointInfoDetailsRequest()
        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")

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

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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.4 Deleting a VPC Endpoint

Function

This API is used to delete a VPC endpoint.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-120 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-121 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

None

Example Requests

Deleting a VPC endpoint

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = DeleteEndpointRequest()
        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")

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

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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.5 Updating a VPC Endpoint

Function

This API is used to update or delete the whitelist of a VPC endpoint.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-122 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-123 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.

Parameter	Mandatory	Type	Description
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-124 Request body parameters

Parameter	Mandatory	Type	Description
whitelist	No	Array of strings	Specifies the whitelist updated or deleted for controlling access to VPC endpoints. The value can be an IPv4 address or CIDR block. The setting descriptions are as follows: ● If the value is specified, the whitelist is updated to the specified value. ● If this parameter is left blank, all whitelists are deleted. The whitelist is left blank by default. Minimum: **0** Maximum: **32**
enable_whitelist	No	Boolean	Specifies whether to enable network ACL isolation. ● true: The network ACL isolation is enabled. ● false: The network ACL isolation is disabled. The default value is false.

Response Parameters

Status code: 200

Table 4-125 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates the VPC endpoint services that are configured by the O&M personnel. You can use them directly without creating them by yourselves. • **interface**: indicates the cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create interface VPC endpoint services.
status	String	Specifies the status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint failed to be created. • **deleting**: The VPC endpoint is being deleted.

Parameter	Type	Description
ip	String	Specifies the IP address for accessing the associated VPC endpoint service. This parameter is returned only under the following conditions: ● You query a VPC endpoint for accessing an interface VPC endpoint service. ● Connection approval is enabled for the VPC endpoint service, and the connection has been approved. **status** of the VPC endpoint can be **accepted** or **rejected**. The **rejected** status only appears when the VPC endpoint is accepted and then rejected. Minimum: **1** Maximum: **64**
active_status	Array of strings	Specifies the account status. ● **frozen**: The account is frozen. ● **active**: The account is unfrozen.
endpoint_service_name	String	Specifies the name of the VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
enable_dns	Boolean	Specifies whether to create a domain name. ● **true**: Create a domain name. ● **false**: Do not create a domain name. Note When you create a VPC endpoint for connecting to a gateway VPC endpoint service, no domain name is created regardless of whether **enable_dns** is set to **true** or **false**.
dns_names	Array of strings	Specifies the domain name for accessing the associated VPC endpoint service. This parameter is available when **enable_dns** is set to **true**. Minimum: **0** Maximum: **32**
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in UUID format. Minimum: **1** Maximum: **64**

Parameter	Type	Description
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see "Obtaining a Project ID". Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, access control is disabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**

Table 4-126 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-127 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

Updating a VPC endpoint (Setting **whitelist** to **192.168.1.1** and **192.168.1.2** and **enable_whitelist** to **true**)

```
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,
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

Updating a VPC endpoint (Setting **whitelist** to **192.168.1.1** and **192.168.1.2** and **enable_whitelist** to **true**)

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

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



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

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

Updating a VPC endpoint (Setting **whitelist** to **192.168.1.1** and **192.168.1.2** and **enable_whitelist** to **true**)

```
# 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 = UpdateEndpointWhiteRequest()
        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

Updating a VPC endpoint (Setting **whitelist** to **192.168.1.1** and **192.168.1.2** and **enable_whitelist** to **true**)

```
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.UpdateEndpointWhiteRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.6 Modifying Route Tables Associated with a VPC Endpoint

Function

This API is used to modify route tables associated with a VPC endpoint.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-128 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-129 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-130 Request body parameters

Parameter	Mandatory	Type	Description
routetables	Yes	Array of strings	Lists the IDs of route tables. Minimum: **0** Maximum: **64** Array Length: **1 - 10**

Response Parameters

Status code: 200

Table 4-131 Response body parameters

Parameter	Type	Description
routetables	Array of strings	Specifies the IDs of route tables. If this parameter is not specified, the ID of the route table of the default VPC is returned. This parameter is available when you update VPC endpoints for connecting to gateway VPC endpoint services. Minimum: **0** Maximum: **64**

Parameter	Type	Description
error	Array of **RoutetableInfoError** objects	Specifies the error message returned when the route table of the VPC endpoint subnet fails to be modified.

Table 4-132 RoutetableInfoError

Parameter	Type	Description
bind_failed	Array of **RoutetableInfoErrorDetial** objects	Failed to bind the route table to the VPC endpoint subnet.
unbind_failed	Array of **RoutetableInfoErrorDetial** objects	Failed to unbind the route table from the VPC endpoint subnet.

Table 4-133 RoutetableInfoErrorDetial

Parameter	Type	Description
id	String	Routing table ID. Minimum: **0** Maximum: **64**
error_message	String	Detailed error information. Minimum: **0** Maximum: **1024**

Example Requests

Modifying route tables associated with a VPC endpoint

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

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

Example Responses

Status code: 200

The server has successfully processed the request.


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

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying route tables associated with a VPC endpoint

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateEndpointRoutetableRequest request = new UpdateEndpointRoutetableRequest();
        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.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

Modifying route tables associated with a VPC endpoint

```
# 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 = UpdateEndpointRoutetableRequest()
        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

Modifying route tables associated with a VPC endpoint

```
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.UpdateEndpointRoutetableRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.7 Modifying the Policy of a Gateway VPC Endpoint

Function

This API is used to modify the policy of a VPC endpoint.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-134 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-135 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-136 Request body parameters

Parameter	Mandatory	Type	Description
policy_statement	No	Array of **PolicyStatement** objects	Only gateway VPC endpoints with both ends fixed are involved.
policy_document	No	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**

Table 4-137 PolicyStatement

Parameter	Mandatory	Type	Description
Effect	Yes	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Yes	Array of strings	Specifies OBS access permissions.
Resource	Yes	Array of strings	Specifies the OBS object.

Response Parameters

Status code: 200

Table 4-138 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates VPC endpoint services that are configured by the O&M personnel. You can directly use them. • **interface**: indicates cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create interface VPC endpoint services.

Parameter	Type	Description
status	String	Specifies the VPC endpoint status. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint service failed to be created. • **deleting**: The VPC endpoint service is being deleted.
active_status	Array of strings	Specifies the account status. • **frozen**: The account is frozen. • **active**: The account is unfrozen.
endpoint_service_name	String	Specifies the name of a VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
ip	String	Specifies the IP address for accessing the associated VPC endpoint service. This parameter is returned only under the following conditions: • You query a VPC endpoint for accessing an interface VPC endpoint service. • Connection approval has been enabled for the VPC endpoint service, and the connection has been approved. **status** of the VPC endpoint can be **accepted** or **rejected**. The **rejected** status only appears when the VPC endpoint is accepted and then rejected. Minimum: **1** Maximum: **64**
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**

Parameter	Type	Description
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see Obtaining a Project ID. Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of **status** is **failed**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, **false** will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Specifies the route table ID list. If this parameter is not specified, the ID of the default route table of the VPC is returned. This parameter is available when you create a VPC endpoint for connecting to a gateway VPC endpoint service. Minimum: **0** Maximum: **64**

Parameter	Type	Description
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**
endpoint_pool_id	String	(To be discarded) Specifies the ID of the cluster associated with the VPC endpoint. Minimum: **1** Maximum: **64**
public_border_group	String	Specifies the information about the public border group associated with the VPC endpoint. This parameter is returned only when the VPC endpoint is associated with an edge pool.

Table 4-139 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**

Parameter	Type	Description
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-140 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Table 4-141 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

- Modifying the policy of a gateway VPC endpoint (Setting **Action** to **obs:*:***, **Resource** to **obs:**:*:*/*** and **obs:**:*:**, and **Effect** to** Allow**)

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

{
  "policy_statement" : [ {
    "Action" : [ "obs:*:*" ],
    "Resource" : [ "obs:*:*:*:*/*", "obs:*:*:*:*" ],
    "Effect" : "Allow"
```



```
 } ]
}
```

- Modifying the policy of a VPC endpoint

```
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" : [ "*" ]
    } ]
  }
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

- ```
  {
    "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 Sample Code

The SDK sample code is as follows.

Java

- Modifying the policy of a gateway VPC endpoint (Setting **Action** to **obs:*:***, **Resource** to **obs:*:*:/*** and **obs:*:*:***, and **Effect** to** Allow**)

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateEndpointPolicyRequest request = new UpdateEndpointPolicyRequest();
        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());
        }
    }
}
```

- Modifying the policy of a VPC endpoint

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

- Modifying the policy of a gateway VPC endpoint (Setting **Action** to **obs:*:***, **Resource** to **obs:*:*:*/*** and **obs:*:*:***, and **Effect** to** Allow**)

```
# 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 = UpdateEndpointPolicyRequest()
        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)
```

- Modifying the policy of a VPC endpoint

```
# 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 = UpdateEndpointPolicyRequest()
        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

- Modifying the policy of a gateway VPC endpoint (Setting **Action** to **obs:*:***, **Resource** to **obs:*:*:/*** and **obs:*:*:***, and **Effect** to** Allow**)

```
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.UpdateEndpointPolicyRequest{}
    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)
    }
}
```

- Modifying the policy of a VPC endpoint

```
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.UpdateEndpointPolicyRequest{}
    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)
    }
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.3.8 Deleting the Policy of a Gateway VPC Endpoint

Function

This API is used to delete the policy of a gateway VPC endpoint. This API will not be used anymore and is not recommended.

Calling Method

For details, see **Calling APIs**.

URI

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

Table 4-142 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID. Minimum: **1** Maximum: **64**
vpc_endpoint_id	Yes	String	Specifies the ID of the VPC endpoint. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-143 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-144 Response body parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint. Minimum: **1** Maximum: **64**
service_type	String	Specifies the type of the VPC endpoint service that the VPC endpoint is used to connect to. • **gateway**: indicates VPC endpoint services that are configured by the O&M personnel. You can directly use them. • **interface**: indicates cloud services configured by the O&M personnel and private services created by yourselves. You cannot configure these cloud services, but can use them. You can query the public VPC endpoint services to view the VPC endpoint services that are visible and accessible to all users and are configured by the O&M personnel. You can create interface VPC endpoint services.

Parameter	Type	Description
status	String	Specifies the VPC endpoint status. • **pendingAcceptance**: The VPC endpoint is to be accepted. • **creating**: The VPC endpoint is being created. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint service failed to be created. • **deleting**: The VPC endpoint service is being deleted.
active_status	Array of strings	Specifies the account status. • **frozen**: The account is frozen. • **active**: The account is unfrozen.
endpoint_service_name	String	Specifies the name of a VPC endpoint service.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service. Minimum: **1** Maximum: **64**
ip	String	Specifies the IP address for accessing the associated VPC endpoint service. This parameter is returned only under the following conditions: • You query a VPC endpoint for accessing an interface VPC endpoint service. • Connection approval has been enabled for the VPC endpoint service, and the connection has been approved. **status** of the VPC endpoint can be **accepted** or **rejected**. The **rejected** status only appears when the VPC endpoint is accepted and then rejected. Minimum: **1** Maximum: **64**
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created. Minimum: **1** Maximum: **64**

Parameter	Type	Description
created_at	String	Specifies when the VPC endpoint was created. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format **YYYY-MM-DDTHH:MM:SSZ** is used.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see Obtaining a Project ID. Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
error	Array of **QueryError** objects	Specifies the error message. This field is returned when the VPC endpoint is abnormal, that is, the value of **status** is **failed**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service. Minimum: **0** Maximum: **32**
enable_whitelist	Boolean	Specifies whether access control is enabled. • **true**: Access control is enabled. • **false**: Access control is disabled. If you do not specify this parameter, **false** will be returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Specifies the route table ID list. If this parameter is not specified, the ID of the default route table of the VPC is returned. This parameter is available when you create a VPC endpoint for connecting to a gateway VPC endpoint service. Minimum: **0** Maximum: **64**

Parameter	Type	Description
description	String	Specifies the description field. The value can contain characters such as letters and digits, but cannot contain less than signs (<) nor great than signs (>). Minimum: **0** Maximum: **512**
policy_statement	Array of **PolicyStatement** objects	Specifies the policy of the gateway VPC endpoint. This parameter is only available when **enable_policy** of the VPC endpoint services for OBS and SFS is set to **true**. Array Length: **0 - 10**
policy_document	Object	Specifies the IAM 5.0 policies. Array Length: **0 - 20480**
endpoint_pool_id	String	(To be discarded) Specifies the ID of the cluster associated with the VPC endpoint. Minimum: **1** Maximum: **64**
public_border_group	String	Specifies the information about the public border group associated with the VPC endpoint. This parameter is returned only when the VPC endpoint is associated with an edge pool.

Table 4-145 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**

Parameter	Type	Description
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Table 4-146 QueryError

Parameter	Type	Description
error_code	String	Error code. Minimum: **0** Maximum: **10**
error_message	String	Error message. Minimum: **0** Maximum: **1024**

Table 4-147 PolicyStatement

Parameter	Type	Description
Effect	String	• **Allow** indicates that the VPC endpoint policy can be modified. • **Deny** indicates that the VPC endpoint policy cannot be modified.
Action	Array of strings	Specifies OBS access permissions.
Resource	Array of strings	Specifies the OBS object.

Example Requests

Deleting the policy of a gateway VPC endpoint

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

Example Responses

Status code: 200

The server has successfully processed the request.

```
{
  "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 Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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 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");

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

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

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```



```
        WithSk(sk).
        Build()

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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.4 Resource Quotas

4.4.1 Query Quotas

Function

This API is used to query the quota of your resources, including the quota of VPC endpoint services and the quota of VPC endpoints.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/quotas

Table 4-148 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**

Table 4-149 Query Parameters

Parameter	Mandatory	Type	Description
type	No	String	Specifies the resource type. • endpoint_service • endpoint

Request Parameters

Table 4-150 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-151 Response body parameters

Parameter	Type	Description
quotas	**ResourcesResponseBody** object	Specifies the response body of the API for querying quotas.

Table 4-152 ResourcesResponseBody

Parameter	Type	Description
resources	Array of **Quotas** objects	Specifies the resource list.

Table 4-153 Quotas

Parameter	Type	Description
type	String	Specifies the resource type. You can query the quota of a specified resource by resource type. ● endpoint_service ● endpoint
used	Integer	Specifies the number of created resources. The value ranges from 0 to the value of quota. Minimum: **0**
quota	Integer	Specifies the resource quota. The value ranges from the default quota value to the maximum quota value. Minimum: **0**

Example Requests

- Querying the VPC endpoint service quotas

  ```
  GET https://{endpoint}/v1/{project_id}/quotas?type=endpoint_service
  ```

- Querying the VPC endpoint quotas

  ```
  GET https://{endpoint}/v1/{project_id}/quotas?type=endpoint
  ```

Example Responses

Status code: 200

The server has successfully processed the request.

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


SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        ListQuotaDetailsRequest request = new ListQuotaDetailsRequest();
        request.withType(ListQuotaDetailsRequest.TypeEnum.fromValue("<type>"));
        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"]

    credentials = BasicCredentials(ak, sk)

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

    try:
        request = ListQuotaDetailsRequest()
        request.type = "<type>"
        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")

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

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

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


More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.5 Tags

4.5.1 Querying Resources by Tag

Function

This API is used to query resources of a tenant by tag.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{resource_type}/resource_instances/action

Table 4-154 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
resource_type	Yes	String	Specifies the resource type. The type is endpoint_service or endpoint.

Request Parameters

Table 4-155 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-156 Request body parameters

Parameter	Mandatory	Type	Description
tags	No	Array of **TagValuesList** objects	Tags are included. A maximum of 10 tag keys are included, and each tag value can have a maximum of 10 values. Each tag value can be an empty array, but the tag structure cannot be missing. Tag keys must be unique. Values of the same tag key must be unique. Specifies all returned tags of the resource. Keys are in the AND relationship while values in the key-value structure are in the OR relationship. If no filtering condition for tags is specified, full data is returned. Array Length: **1 - 10**

Parameter	Mandatory	Type	Description
tags_any	No	Array of **TagValuesList** objects	Any tags are included. A maximum of 10 tag keys are included, and each tag key can have a maximum of 10 values. Values of each tag key can be an empty array, but the tag structure cannot be missing. Both tag keys and values of the same tag key must be unique. Specifies resources including tags returned. Keys and values in the key-value structure are in the OR relationship. If no filtering condition is specified, full data is returned. Array Length: **1 - 10**
not_tags	No	Array of **TagValuesList** objects	Tags are excluded, a maximum of 10 tag keys are included, and each tag value can have a maximum of 10 values. Each tag value can be an empty array, but the tag structure cannot be missing. Tag keys must be unique. Values of the same tag key must be unique. Specifies resources excluding tags returned. Keys are in the AND relationship while values in the key-value structure are in the OR relationship. If no filtering condition is specified, full data is returned. Array Length: **1 - 10**

Parameter	Mandatory	Type	Description
not_tags_any	No	Array of **TagValuesList** objects	Any tags are excluded, a maximum of 10 tag keys are included, and each tag value can have a maximum of 10 values. Each tag value can be an empty array, but the tag structure cannot be missing. Tag keys must be unique. Values of the same tag key must be unique. Specifies resources excluding tags returned. Keys are in the AND relationship while values in the key-value structure are in the OR relationship. If no filtering condition is specified, full data is returned. Array Length: **1 - 10**
sys_tags	No	Array of **TagValuesList** objects	Specifies the system tags.
limit	No	String	Specifies the number of records to be queried. This parameter is unavailable when action is set to count. If action is set to filter, the default value is 1000. The maximum value of limit is 1000 and the minimum value is 1.
offset	No	String	Specifies the index position and offset. The query starts from the data after the first data offset. This parameter is unavailable when action is set to count. If action is set to filter, the default value is 0 which indicates that the query starts from the first data. The value must be a non-negative number.

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies the action, which only can be filter and count. If this parameter is set to filter, the query is performed based on the filter criteria. If the parameter is set to count, only the total number of resources need to be returned.
matches	No	Array of **Match** objects	Specifies the search field. The tag key is the field to be matched, for example, resource_name. The tag value indicates the matched value. The tag key is a fixed dictionary value and cannot be a duplicate key or an unsupported key. Specifies whether fuzzy match is required based on the value of the tag key. For example, if the tag key is resource_name, fuzzy search (case-insensitive) is used by default. If value is an empty string, exact match is used. An empty list is returned because the resource_name for most services cannot be left blank. If the tag key is resource_id, exact match is used. Only resource_name for key is supported. Other key values will be available later.
without_any_tag	No	Boolean	The default value is false. The value can be true or false. When withoutAnyTag is set to true, ignore parameter verification on the tags, tagsAny, notTags, and notTagsAny.

Table 4-157 TagValuesList

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It contains a maximum of 127 Unicode characters. A tag key cannot be left blank. (The character set of this parameter is not verified during the search.) The tag key cannot be left blank, empty strings, or spaces. Before you verify and use it, delete spaces at the beginning and end of the tag key. Minimum: **1** Maximum: **128**
values	Yes	Array of strings	Specifies the tag values. Each tag value contains a maximum of 255 Unicode characters. Before you verify and use it, delete spaces at the beginning and end of the tag value. The tag value can be an empty array but cannot be left blank. If values is left blank, it indicates any_value (querying any value). Values are in the OR relationship. The system does not verify the character set of values when searching for resources, but only verifies the length. Maximum: **255** Array Length: **1 - 1000**

Table 4-158 Match

Parameter	Mandatory	Type	Description
key	Yes	String	Indicates the tag key. Only resource_name for key is supported. Other key values will be available later. Minimum: **1** Maximum: **36**

Parameter	Mandatory	Type	Description
value	Yes	String	Value. Each value can contain a maximum of 255 Unicode characters. The character set of value is not verified. Minimum: **1** Maximum: **43**

Response Parameters

Status code: 200

Table 4-159 Response body parameters

Parameter	Type	Description
resources	Array of **ResourceInstance** objects	Specifies the resource details.
total_count	Integer	Specifies the total number of records. Minimum: **1**

Table 4-160 ResourceInstance

Parameter	Type	Description
resource_id	String	Indicates the resource ID, endpoint service ID, or endpoint ID. Minimum: **1** Maximum: **64**
tags	Array of **TagList** objects	Indicates the list of tags.
resource_name	String	Resource name. If the resource does not have a name, the ID is returned. Minimum: **0** Maximum: **128**

Table 4-161 TagList

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. It cannot be left blank. It cannot contain equal signs (=), asterisks (*), less than signs (<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Minimum: **1** Maximum: **128**
value	String	Specifies the tag key. A tag value contains a maximum of 43 Unicode characters and can be an empty string. It cannot contain equal signs (=), asterisks (*), less than signs(<), greater than signs (>), backslashes (), commas (,), vertical bars (|), and slashes (/), and the first and last characters cannot be spaces. Maximum: **255**

Example Requests

- Querying VPC endpoints by tag

```
POST https://{endpoint}/v1/{project_id}/endpoint/resource_instances/action

When action is set to filter:    "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"
        ]
    }
  ]
}
```

- Querying the total number of VPC endpoint services

```
https://{endpoint}/v1/{project_id}/endpoint_service/ resource_instances/action

When action is set to count:
{
  "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"
    }
```



```
    ]
}
```

Example Responses

Status code: 200

The server has successfully processed the request.

```
– When **action** is set to **filter**:
{
"resources": [
{
"resource_id": "cdfs_cefs_wesas_12_dsad",
"resource_name": "resource1",
"tags": [
{
"key": "key1", "value": "value1"
},
{
"key": "key2", "value": "value1"
}
]
}
],
"total_count": 1000
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Querying VPC endpoints by tag

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

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

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

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



```
            .build();
        ListResourceInstancesRequest request = new ListResourceInstancesRequest();
        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());
        }
    }
}
```

- Querying the total number of VPC endpoint services

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

- Querying VPC endpoints by tag

```
# 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 = ListResourceInstancesRequest()
        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)
```

- Querying the total number of VPC endpoint services

```
# 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 = ListResourceInstancesRequest()
        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

- Querying VPC endpoints by tag

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

- Querying the total number of VPC endpoint services

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.5.2 Batch Adding or Deleting Tags to or from a Resource

Function

This API is used to batch add or delete tags to and from a specified VPC endpoint service or VPC endpoint.

- A resource can have up to 10 tags.

Calling Method

For details, see **Calling APIs**.

URI

POST /v1/{project_id}/{resource_type}/{resource_id}/tags/action

Table 4-162 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
resource_type	Yes	String	Specifies the resource type. The type is endpoint_service or endpoint.
resource_id	Yes	String	Indicates the resource ID, endpoint service ID, or endpoint ID. Minimum: **1** Maximum: **64**

Request Parameters

Table 4-163 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Table 4-164 Request body parameters

Parameter	Mandatory	Type	Description
tags	No	Array of **ResourceTag** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies the operation to be performed, which only can be • **create** • **delete**

Table 4-165 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. A key can contain up to 128 Unicode characters. The key must comply with the tag key configuration requirements. Minimum: **1** Maximum: **128**
value	No	String	Specifies the tag value. **value** is mandatory when **action** is set to **create**. Each value can contain a maximum of 255 Unicode characters. If **value** is specified, tags are deleted by key and value. If **value** is not specified, tags are deleted by key. The tag value must comply with the tag value configuration requirements. Maximum: **255**

Response Parameters

None

Example Requests

- Adding resource tags

```
POST https://{endpoint}/v1/{project_id}/endpoint_service/{resource_id}/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key",
    "value" : "value3"
  } ]
}
```


- Deleting resource tags

```
POST https://{endpoint}/v1/{project_id}/endpoint/{resource_id}/tags/action

{
  "action" : "delete",
  "tags" : [ {
    "key" : "key1"
  }, {
    "key" : "key2",
    "value" : "value3"
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

- Adding resource tags

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

        VpcepClient client = VpcepClient.newBuilder()
                .withCredential(auth)
                .withRegion(VpcepRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchAddOrRemoveResourceInstanceRequest request = new BatchAddOrRemoveResourceInstanceRequest();
        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());
        }
    }
}
```

- Deleting resource tags

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

- Adding resource tags

```
# 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 = BatchAddOrRemoveResourceInstanceRequest()
        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)
```

- Deleting resource tags

```
# 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 = BatchAddOrRemoveResourceInstanceRequest()
        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

- Adding resource tags

```
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.BatchAddOrRemoveResourceInstanceRequest{}
    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)
    }
}
```

- Deleting resource tags

```
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.BatchAddOrRemoveResourceInstanceRequest{}
    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)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The server has successfully processed the request.

Error Codes

See **Error Codes**.

4.5.3 Querying Resource Tags

Function

This API is used to query tags of resources of a tenant based on the tenant ID and resource type.

Calling Method

For details, see **Calling APIs**.

URI

GET /v1/{project_id}/{resource_type}/tags

Table 4-166 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Minimum: **1** Maximum: **64**
resource_type	Yes	String	Specifies the resource type. The type is endpoint_service or endpoint.

Request Parameters

Table 4-167 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. It can be obtained by calling the IAM API. The value of X-Subject-Token in the response header is the user token.
Content-Type	No	String	Specifies the MIME type of the request body. Default value application/json is recommended. For APIs used to upload objects or images, the MIME type varies depending on the flow type. Default: **application/json**

Response Parameters

Status code: 200

Table 4-168 Response body parameters

Parameter	Type	Description
tags	Array of TagValuesList objects	Tags are included. A maximum of 10 tag keys are included, and each tag value can have a maximum of 10 values. Each tag value can be an empty array, but the tag structure cannot be missing. Tag keys must be unique. Values of the same tag key must be unique. Specifies all returned tags of the resource. Keys are in the AND relationship while values in the key-value structure are in the OR relationship. If no filtering condition for tags is specified, full data is returned. Array Length: **1 - 10**

Table 4-169 TagValuesList

Parameter	Type	Description
key	String	Specifies the tag key. It contains a maximum of 127 Unicode characters. A tag key cannot be left blank. (The character set of this parameter is not verified during the search.) The tag key cannot be left blank, empty strings, or spaces. Before you verify and use it, delete spaces at the beginning and end of the tag key. Minimum: **1** Maximum: **128**
values	Array of strings	Specifies the tag values. Each tag value contains a maximum of 255 Unicode characters. Before you verify and use it, delete spaces at the beginning and end of the tag value. The tag value can be an empty array but cannot be left blank. If values is left blank, it indicates any_value (querying any value). Values are in the OR relationship. The system does not verify the character set of values when searching for resources, but only verifies the length. Maximum: **255** Array Length: **1 - 1000**

Example Requests

- Querying tags of a VPC endpoint service

  ```
  GET https://{endpoint}/v1/{project_id}/endpoint_service/tags
  ```


- Querying tags of a VPC endpoint

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

Example Responses

Status code: 200

Indicates the list of tags.

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

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import 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");

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

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

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

    client := vpcep.NewVpcepClient(
        vpcep.VpcepClientBuilder().
```



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

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

More

For SDK sample code of more programming languages, see the Sample Code tab in **API Explorer**. SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Indicates the list of tags.

Error Codes

See **Error Codes**.

5 Application Examples

5.1 Configuring a VPC Endpoint for Communication Across VPCs

Scenarios

VPC Endpoint enables you to privately connect your VPC to a VPC endpoint service (a cloud service or your private service) in another VPC, providing higher access efficiency and networking security compared with EIPs.

This section describes how to invoke APIs described in **Creating a VPC Endpoint Service** and **Creating a VPC Endpoint** to connect a VPC endpoint to a VPC endpoint service. For details, see **Calling APIs**.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use one token for authentication, you can cache it to avoid frequently calling the IAM API.

Prerequisites

You have planned the region where you want to create a VPC endpoint and obtained the endpoint required for API calls. For details, see **Endpoints**.

Creating a VPC Endpoint Service

The following is an example request of creating an interface VPC endpoint service for an ECS:

NOTE

Before creating a VPC endpoint service, obtain necessary information such as values of parameters **vpc_id** and **port_id**. For details, see **Creating a VPC Endpoint Service**.

```
{
  "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**: indicates the ID for identifying the backend service of a VPC endpoint service. For example, when you create a VPC endpoint service for an ECS, set this parameter to the NIC ID of the ECS IP address.
- **vpc_id**: indicates the ID of the VPC where the backend resource is located.
- **approval_enabled**: indicates whether approval is required when a VPC endpoint connects to a VPC endpoint service. For example, if this parameter is set to **false**, no approval is required.
- **service_type**: indicates the type of the VPC endpoint service. For example, if this parameter is set to **interface**, the created VPC endpoint service is an interface VPC endpoint service.
- **server_type**: indicates the type of the backend resource. For example, if this parameter is set to **VM**, the backend resource is an ECS.
- **ports.client_port**: indicates the port provided by the VPC endpoint, allowing you to access the VPC endpoint service.
- **ports.server_port**: indicates the port provided by the backend resource to provide services.
- **ports.protocol**: indicates the protocol used for the port mapping.

Creating a VPC Endpoint

The following is an example request for creating a VPC endpoint, and creating a private domain name is supported.

NOTE

Before creating a VPC endpoint, obtain necessary information, such as values of parameters **vpc_id** and **subnet_id** and the VPC endpoint service ID returned in the previous step. For details, see **Creating a VPC Endpoint**.

```
{
 "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**: indicates the ID of the subnet where the VPC endpoint is created.
- **vpc_id**: indicates the ID of the VPC where the VPC endpoint is created.
- **endpoint_service_id**: indicates the ID of the VPC endpoint service.
- **enable_dns**: indicates whether to create a private domain name for the VPC endpoint. For example, if this parameter is set to **true**, a private domain name is created for the VPC endpoint.

After the VPC endpoint is created, instances in the VPC where the VPC endpoint is located can use the VPC endpoint to access the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed83**.

6 Appendix

6.1 Status Code

- Normal

  Table 6-1 Return values for successful requests

Status Code	Returned Value	Description
200	OK	The results of POST, GET, and PUT operations are returned as expected.
204	No Content	The results of the DELETE operation are returned as expected.

- Abnormal

  Table 6-2 Return code for failed requests

Status Code	Returned Value	Description
400	Bad Request	The server failed to process the request.
401	Unauthorized	You must enter a username and password to access the requested page.
403	Forbidden	You are forbidden to access the requested page.
404	Not Found	The server could not find the requested page.
405	Method Not Allowed	You are not allowed to use the method specified in the request.

Status Code	Returned Value	Description
406	Not Acceptable	The response generated by the server could not be accepted by the client.
407	Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.
408	Request Timeout	The request timed out.
409	Conflict	The request could not be processed due to a conflict.
500	Internal Server Error	Failed to complete the request because of an internal service error.
501	Not Implemented	Failed to complete the request because the server does not support the requested function.
502	Bad Gateway	Failed to complete the request because the server has received an invalid response.
503	Service Unavailable	Failed to complete the request because the service is unavailable.
504	Gateway Timeout	A gateway timeout error occurred.

6.2 Error Codes

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **API Gateway Error Codes**.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.0002	Parameter error.	Parameter error.	Check whether the parameter is valid.
400	EndPoint.0006	Invalid limit.	Invalid limit.	Enter a valid limit.
400	EndPoint.0007	Invalid action.	Invalid action.	Enter a valid action.
400	EndPoint.0009	The remote address does not match.	The remote address does not match.	Check whether you have the access permissions.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.0010	Invalid offset.	Invalid offset.	Enter a valid offset.
400	EndPoint.0014	Invalid project ID.	Invalid project ID.	Enter a valid project ID.
400	EndPoint.0015	Invalid specification.	Invalid specifications.	Enter valid specifications.
400	EndPoint.0016	The number of batch operated resources exceeded the limit.	The number of batch operated resources exceeded the limit.	Reduce the number of resources to be batch operated.
400	EndPoint.0017	Invalid sort_key.	Invalid sort key.	Enter a valid sort key.
400	EndPoint.0018	Invalid sort_dir.	Invalid sort DIR.	Enter a valid sort DIR.
400	EndPoint.0019	Invalid status.	Invalid status.	Enter a valid status.
400	EndPoint.0020	Invalid VPC ID.	Invalid VPC ID.	Enter a valid VPC ID.
400	EndPoint.0021	Invalid marker_id.	Invalid marker ID.	Enter a valid marker ID.
400	EndPoint.0022	The number of requests exceeded the limit. Please try later.	The number of requests exceeded the limit.	Try again later.
400	EndPoint.0023	Invalid subnet_id.	Invalid subnet ID.	Enter a valid subnet ID.
400	EndPoint.1003	Invalid service name.	Invalid service name.	Enter a valid service name.
400	EndPoint.1004	Invalid request.	Invalid request.	Enter a valid request body.
400	EndPoint.1005	DNS service Failed: Failed to create a DNS zone.	DNS failure: Failed to create the DNS zone.	Contact technical support.
400	EndPoint.1008	Failed to obtain the token.	Failed to obtain the token.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	Endpoint.1018	Quota exceeded.	Quota exceeded.	Contact technical support.
400	EndPoint.1019	Invalid route table ID.	Invalid route table ID.	Enter a valid route table ID.
400	EndPoint.2001	The VPC does not exist.	The VPC does not exist.	Enter a valid VPC ID for the current tenant.
400	EndPoint.2002	The request input parameter is empty.	The request input parameter is left blank.	Enter a valid parameter.
400	EndPoint.2003	The endpoint service does not exist.	The endpoint service does not exist.	Enter a valid VPC endpoint service.
400	EndPoint.2004	The endpoint service is unavailable.	The VPC endpoint service is unavailable.	Try again later. If the fault persists, contact technical support.
400	EndPoint.2009	The specification information does not exist.	The specifications information does not exist.	Enter correct specifications.
400	EndPoint.2010	The input parameter subnet ID is empty.	The input parameter **subnet_id** is left blank.	Enter a valid subnet ID.
400	EndPoint.2011	The input parameter VPC ID is empty.	The input parameter **vpc_id** is left blank.	Enter a valid VPC ID.
400	EndPoint.2012	You have no permission to connect to the VPC endpoint service.	You have no permission to connect to the VPC endpoint service.	Check whether you have the access permissions.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2013	The endpoint does not belong to the endpoint service.	The VPC endpoint does not belong to the VPC endpoint service.	Check whether the VPC endpoint is valid.
400	EndPoint.2014	The endpoint has connected to the endpoint service.	The VPC endpoint has connected to the VPC endpoint service.	Connected. You do not need to connect again.
400	EndPoint.2015	The endpoint has been frozen.	The VPC endpoint has been frozen.	Contact technical support.
400	EndPoint.2016	The endpoint pool IP address does not exist.	The pool IP address of the VPC endpoint does not exist.	Contact technical support.
400	EndPoint.2017	Invalid endpoint ID.	Invalid VPC endpoint ID.	Enter a valid VPC endpoint ID.
400	EndPoint.2018	The endpoint is being deleted.	The VPC endpoint is being deleted.	Select an available VPC endpoint.
400	EndPoint.2019	The endpoint is being created.	The VPC endpoint is being created.	Try again later.
400	EndPoint.2020	qrMac or sgMac does not exist.	**qrMac** or **sgMac** is not found.	Contact technical support.
400	EndPoint.2021	Failed to query the VPC.	Failed to query the VPC.	Contact technical support.
400	EndPoint.2022	Failed to create an endpoint.	Failed to create the VPC endpoint.	Contact technical support.
400	EndPoint.2023	CIDR is not found.	CIDR is not found.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2024	shadowVpc or shadowPort does not exist.	**shadowVpc** or **shadowPort** is left blank.	Contact technical support.
400	EndPoint.2025	The endpoint port does not exist.	The VPC endpoint port is not found.	Contact technical support.
400	EndPoint.2026	VNI is empty.	VNI is left blank.	Contact technical support.
400	EndPoint.2027	Invalid action.	Invalid action.	Enter a valid action.
400	EndPoint.2028	The endpoint service port or protocol is empty.	The VPC endpoint service port or protocol is left blank.	Enter a valid port number or protocol.
400	EndPoint.2029	The requested endpoint service ID is empty.	The requested endpoint service ID is left blank.	Enter a valid VPC endpoint service ID.
400	EndPoint.2031	Only one endpoint is allowed.	Only one VPC endpoint is allowed.	A VPC endpoint service can accept or reject only one VPC endpoint.
400	EndPoint.2033	The entered parameter enable_dns is invalid.	Invalid **enable_dns** value.	Enter a valid **enable_dns**.
400	EndPoint.2034	The entered parameter enable_dns is incorrect.	Incorrect **enable_dns** value.	Enter a correct **enable_dns** value.
400	EndPoint.2035	The system parameter dns.enable is invalid.	Invalid **dns.enable** value.	Contact technical support.
400	EndPoint.2037	The current network does not belong to the VPC.	The current network does not belong to the VPC.	Check whether the parameter is valid.
400	EndPoint.2038	The pool does not exist.	The pool does not exist.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2039	The route table is being used by another VPC endpoint.	The route table is being used by another VPC endpoint.	Contact technical support.
400	EndPoint.2040	The VPC endpoint has no route table bound.	The VPC endpoint has no route table associated.	Contact technical support.
400	EndPoint.2041	Invalid Port IP address.	Invalid NIC IP address.	Enter a valid IP address.
400	EndPoint.2042	The Port IP is in use.	The IP address of the NIC is in use.	Enter another IP address.
400	EndPoint.2043	The Port IP is not valid for the subnet.	The IP address of the NIC does not belong to the subnet.	Enter a valid IP address.
400	EndPoint.2044	Invalid whitelist.	Invalid whitelist.	Enter a valid whitelist.
400	EndPoint.2045	The maximum number of whitelist records has been reached.	The maximum number of records in the whitelist has been reached.	Contact technical support.
400	EndPoint.2046	Can not modify a specified mac endpoint.	**mac** of a specified VPC endpoint cannot be modified.	Contact technical support.
400	EndPoint.2047	The VPC endpoint in the Failed status cannot be modified.	A VPC endpoint in the **Failed** state cannot be modified.	Select a valid VPC endpoint.
400	EndPoint.2048	The VPC endpoint policy is invalid.	Invalid VPC endpoint policy.	Enter a valid VPC endpoint policy.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2049	The VPC endpoint has no policy associated.	No policy has been associated with the VPC endpoint.	Contact technical support.
400	EndPoint.2050	The endpoint has been disabled.	The VPC endpoint has been disabled.	Contact technical support.
400	EndPoint.2051	The current endpoint is switching to new pool.	The VPC endpoint is being switched to a new resource pool.	Try again later.
400	EndPoint.2052	The current endpoint does not need to be rolled back.	The VPC endpoint failed to be rolled back.	Contact technical support.
400	EndPoint.3001	Failed to create a port.	Failed to create the port.	Contact technical support.
400	EndPoint.3002	Invalid permission.	Invalid permissions.	Contact technical support.
400	EndPoint.3003	Invalid port ID.	Invalid port ID.	Contact technical support.
400	EndPoint.3004	Invalid port.	Invalid port.	Contact technical support.
400	EndPoint.3005	Failed to delete the endpoint service.	Failed to delete the VPC endpoint service.	Contact technical support.
400	EndPoint.3006	The endpoint service is being used.	The VPC endpoint service is being used.	Contact technical support.
400	EndPoint.3008	The port does not exist.	The port is not found.	Contact technical support.
400	EndPoint.3009	Invalid CIDR.	Invalid CIDR.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3010	Invalid IP address.	Invalid IP address.	Enter a valid IP address.
400	EndPoint.3011	Parameter IP is not required to create an endpoint service (interface).	**IP** is not required to create an interface VPC endpoint service.	Enter a valid request body.
400	EndPoint.3013	endpointService interface vlan can't have vpcId.	The request for accessing the VLAN VPC endpoint service cannot contain **vpcId**.	Enter a valid request body.
400	EndPoint.3014	endpointService interface can't have cidr.	The request for accessing the VPC endpoint service (interface) cannot contain **cidr**.	Enter a valid request body.
400	EndPoint.3015	endpointService gateway vlan can't have portId.	The request for accessing the VLAN VPC endpoint service (gateway) cannot contain **portId**.	Enter a valid request body.
400	EndPoint.3016	endpointService gateway vlan can't have ip.	The request for accessing the VLAN VPC endpoint service (gateway) cannot contain **ip**.	Enter a valid request body.
400	EndPoint.3017	Invalid CIDRs.	Invalid CIDRs.	Enter valid CIDRs.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3018	endpointService gateway vlan can't have vpcId.	The request for accessing the VLAN VPC endpoint service cannot contain **vpcId**.	Enter a valid request body.
400	EndPoint.3021	Invalid serverType.	Invalid **serverType**.	Enter a valid **serverType**.
400	EndPoint.3022	Failed to create a network.	Failed to create the network.	Contact technical support.
400	EndPoint.3023	Failed to create a subnet.	Failed to create the subnet.	Contact technical support.
400	EndPoint.3035	Invalid action.	Invalid action.	Enter a valid action.
400	EndPoint.3036	Invalid permissions.	The permission list cannot be left blank.	Enter a valid request body.
400	EndPoint.3040	Failed to add a rollback task.	Failed to add the rollback task.	Contact technical support.
400	EndPoint.3042	The port ID does not belong to the current VPC.	The port ID does not belong to the current VPC.	Enter a valid request body.
400	EndPoint.3043	The service port is invalid.	Invalid service port.	Enter a valid request body.
400	EndPoint.3044	The parameter ports conflicted with ports in an existing endpoint service.	This port conflicted with the port of an existing endpoint service.	Enter a valid request body.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3045	Other properties cannot be modified in the current endpoint service state.	Modifying other properties of the VPC endpoint service in the current state is not supported.	Enter a valid request body.
400	EndPoint.3046	The IP address conflicted with an existing endpoint service.	The IP address conflicted with that of an existing VPC endpoint service.	Enter a valid request body.
400	EndPoint.3048	Invalid netType.	Invalid **netType**.	Enter a valid **netType**.
400	EndPoint.3049	The maximum number of whitelist records has been reached.	The maximum number of whitelist records has been reached.	Delete invalid whitelist records or add an authorized account ID named *.
400	EndPoint.3051	Endpoint service vip port id is invalid.	Invalid **vip port id**.	Enter a valid **vip port id**.
400	EndPoint.3052	portId and ip cannot be modified at the same time.	**portId** and **ip** cannot be modified at the same time.	Enter a valid request body.
400	EndPoint.3053	vipPortId and ip cannot be modified at the same time.	**vipPortId** and **ip** cannot be modified at the same time.	Enter a valid request body.
400	EndPoint.3054	portId or vipPortId cannot be modified.	**portId** or **vipPortId** cannot be modified.	Enter a valid request body.
400	EndPoint.3055	ip cannot be modified.	**ip** cannot be modified.	Enter a valid request body.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3056	The maximum of VPC endpoint services using the same IP address has been reached.	The maximum number of VPC endpoint services using the same IP address has been reached.	Contact technical support.
400	EndPoint.3057	cidr cannot be modified.	**cidr** cannot be modified.	Enter a valid request body.
400	EndPoint.3058	The domain name is invalid.	Invalid **dns name**.	Enter a valid **dns name**.
400	EndPoint.3059	The domain name already exists.	The domain name already exists.	Contact technical support.
400	EndPoint.3060	You have no permission to add domain names.	You have no permission to add domain names.	Contact technical support.
400	EndPoint.3061	The maximum number of domain names has reached.	The maximum number of domain names has been reached.	Contact technical support.
400	EndPoint.3062	Invalid endpoint service ID.	Invalid VPC endpoint service ID.	Enter a valid VPC endpoint service ID.
400	EndPoint.3063	Invalid port ID.	Invalid port ID.	Enter a valid port ID.
400	EndPoint.3066	The tag cannot be empty.	The tag cannot be left blank.	Enter a valid request body.
400	EndPoint.3067	The tag key cannot be duplicated.	The tag key must be unique.	Enter a valid request body.
400	EndPoint.3068	Tag keys and values should meet relevant requirements.	Tag keys and values must meet relevant requirements.	Enter a valid request body.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3069	The maximum number of tags has been reached.	The maximum number of tags has been reached.	Contact technical support.
400	EndPoint.3070	Invalid resource type.	Incorrect resource type.	Contact technical support.
400	EndPoint.3071	The tag value cannot be duplicated.	Tag values must be unique.	Contact technical support.
400	EndPoint.3072	The tag key size is invalid.	Invalid tag key length.	Enter a valid tag key.
400	EndPoint.3073	The tag value size is invalid.	Invalid tag value length.	Enter a valid tag value.
400	EndPoint.3074	The maximum of ports has been reached.	The maximum number of port mappings has been reached.	Contact technical support.
400	EndPoint.3075	The protocol is invalid.	Invalid protocol.	Contact technical support.
400	EndPoint.3076	Invalid service name.	Invalid service name.	Enter a valid service name.
400	EndPoint.4001	Failed to query the subnet.	Failed to query the subnet.	Contact technical support.
400	EndPoint.4002	Failed to create a subnet.	Failed to create the subnet.	Contact technical support.
400	EndPoint.4003	Failed to delete the subnet.	Failed to delete the subnet.	Contact technical support.
400	EndPoint.4005	Failed to query the network.	Failed to query the network.	Contact technical support.
400	EndPoint.4006	Failed to create a network.	Failed to create the network.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.4007	Failed to delete the network.	Failed to delete the network.	Contact technical support.
400	EndPoint.4009	Failed to query the port.	Failed to query the port.	Contact technical support.
400	EndPoint.4010	Failed to create a port.	Failed to create the port.	Contact technical support.
400	EndPoint.4011	Failed to delete the port.	Failed to delete the port.	Contact technical support.
400	EndPoint.4013	Failed to query the proxy.	Failed to query the proxy.	Contact technical support.
400	EndPoint.4014	Failed to query the router.	Failed to query the route.	Contact technical support.
400	EndPoint.4015	The router is not found.	The route is not found.	Contact technical support.
400	EndPoint.4016	Failed to add an interface router.	Failed to add the interface route.	Contact technical support.
400	EndPoint.4017	Failed to delete the interface router.	Failed to delete the interface route.	Contact technical support.
400	EndPoint.4018	Failed to add an extension router.	Failed to add the extended route.	Contact technical support.
400	EndPoint.4019	Failed to delete the extension router.	Failed to delete the extended route.	Contact technical support.
400	EndPoint.4020	Failed to query Neutron L3 Agent.	Failed to query Neutron L3 Agent.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.4025	The specification is being used.	The specifications are being used.	Contact technical support.
400	EndPoint.4026	Failed to query the default route table of the VPC.	Failed to query the default route table of the VPC.	Contact technical support.
400	EndPoint.4027	Failed to query route tables of the VPC.	Failed to query route tables of the VPC.	Contact technical support.
400	EndPoint.4028	Failed to add routes to the VPC's route table.	Failed to add routes to the VPC's route table.	Contact technical support.
400	EndPoint.4029	Failed to remove routes from the VPC's route table.	Failed to remove routes from the VPC's route table.	Contact technical support.
401	EndPoint.0003	Authentication failed or authentication information is invalid.	Authentication failed or authentication information is invalid.	Check whether the permissions are enabled.
403	EndPoint.0004	Authentication information is incorrect or you have no permissions.	Authentication information is incorrect or you have no permissions.	Check whether the permissions are enabled.
404	EndPoint.0005	The requested resource is unavailable.	The requested resource is unavailable.	Check whether the input parameters are valid.
404	EndPoint.2006	The requested endpoint does not exist.	The requested VPC endpoint does not exist.	Enter a valid VPC endpoint.

Status Code	Error Code	Error Message	Description	Solution
404	EndPoint.2007	The endpoint information does not exist.	The VPC endpoint information does not exist.	Enter a valid VPC endpoint and check whether the VPC endpoint is deleted.
404	EndPoint.2008	The endpoint has been deleted.	The VPC endpoint has been deleted.	Check whether the VPC endpoint is deleted.
404	EndPoint.2030	markerId is empty.	**markerId** is left blank.	Enter a valid marker ID.
404	EndPoint.4004	The subnet is not found.	The subnet does not exist.	Check the entered subnet ID. If the fault persists, contact technical support.
404	EndPoint.4008	Network is unavailable.	Network is unavailable.	Contact technical support.
404	EndPoint.4012	The port is not found.	The port does not exist.	Contact technical support.
404	EndPoint.4021	Neutron L3 Agent is not found.	Neutron L3 Agent is not found.	Contact technical support.
404	EndPoint.4030	The route table is not found.	The route table does not exist.	Contact technical support.
500	EndPoint.0001	System error. Please retry.	System error. Please retry.	Try again. If the fault persists, contact technical support.
504	EndPoint.0011	The request body is null.	The request body is left blank.	Enter a valid request body.
504	EndPoint.0012	The request header is null.	The request header is left blank.	Enter a request header.
504	EndPoint.0013	The request timed out.	The request timed out.	Contact technical support if the problem persists after a retry.

6.3 Obtaining a Project ID

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- **Obtain the Project ID by Calling an API**
- **Obtain the Project ID from the Console**

Obtain the Project ID by Calling an API

You can obtain a project ID by calling the API used to **query projects based on specified criteria**.

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. {Endpoint} is the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

The following is an example response. The value of **id** is the project 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"
    }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.
   On the **API Credentials** page, view the project ID in the project list.

Figure 6-1 Viewing the project ID