VPC Endpoint

API Reference

Issue 01

Date 2022-04-12

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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.

Contents

1 Before You Start

1.1 Overview

Welcome to *VPC Endpoint API Reference*. The VPC Endpoint (VPCEP) 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 VPCEP resources, such as creating, deleting, or modifying VPC endpoints and VPC endpoint services. For details about all supported operations, see **API Overview**.

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

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 quotas, see **Managing Quotas** in the *VPC Endpoint User Guide*.
- For detailed constraints, see the constraints described in specific APIs.

1.5 Concepts

- Account

  An account is created upon successful registration. 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

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

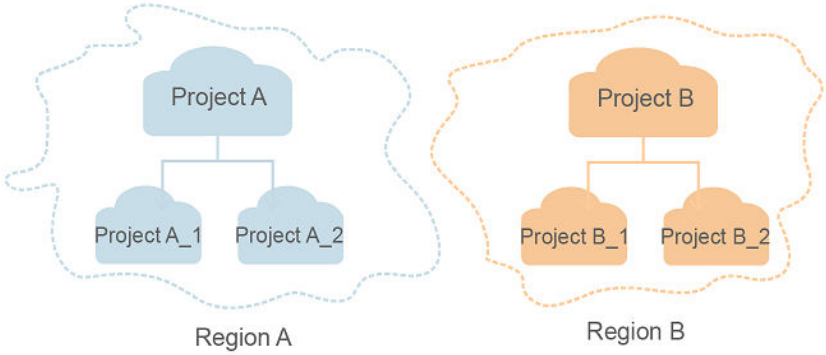
- AZ

  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, deleting, modifying, or querying a VPC endpoint service, querying, adding, or deleting a whitelist record, and querying the VPC endpoint services, the whitelist records, and 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, deleting, or querying a VPC endpoint and 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 **obtaining a user token** 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

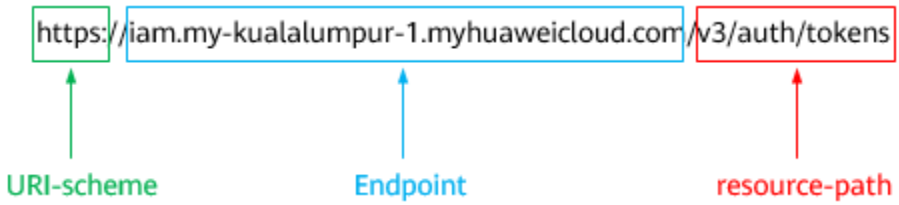
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in the **my-kualalumpur-1** region is **iam.my-kualalumpur-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.

For example, to obtain an IAM token in the **AP-Kuala Lumpur-OP6** region, obtain the endpoint of IAM (**iam.my-kualalumpur-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.my-kualalumpur-1.myhuaweicloud.com/v3/auth/tokens
```

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 used to **obtain a user token**, the request method is **POST**. The request is as follows:

```
POST https://iam.my-kualalumpur-1.myhuaweicloud.com/v3/auth/tokens
```

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	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 API used to **obtain a user token** does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

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

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******** (login password), and *xxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from **Regions and Endpoints**.

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see **Obtaining a User Token**.

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

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

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:

- Token authentication: Requests are authenticated using tokens.
- 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

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.

A cloud service can be deployed as either a project-level service or global service.

- For a project-level service, you need to obtain a project-level token. When you call the API, set **auth.scope** in the request body to **project**.
- For a global service, you need to obtain a global token. When you call the API, set **auth.scope** in the request body to **domain**.

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",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

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

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 .

NOTE

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

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 **obtain a user token**, 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 **obtain a user token**. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-2 Header fields of the response to the request for obtaining a user token

(Optional) Response Body

The body of a response is often returned in structured format 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 **obtain a user token**.

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

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

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

4 APIs

4.1 Version Management

4.1.1 Querying Versions of VPCEP APIs

Function

This API is used to query versions of VPCEP APIs.

URI

GET /

Request

- Example request
  GET https://{endpoint}/

Response

- Parameters

  Table 4-1 Response parameters

Parameter	Type	Description
versions	Array of objects	Lists the versions of VPCEP APIs. For details, see **Table 4-2**.

Table 4-2 VersionModel parameters

Parameter	Type	Description
status	String	Specifies the version status. • **CURRENT**: indicates a major version. • **SUPPORT**: indicates an earlier version which is still supported. • **DEPRECATED**: indicates a deprecated version that may be deleted later.
id	String	Specifies the version ID.
updated	String	Specifies the time when the API version was released. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
version	String	Specifies the supported version.
min_version	String	Specifies the microversion number. If the APIs do not support microversions, the value is left blank.
links	Array of objects	Specifies the API URL. For details, see **Table 4-3**.

Table 4-3 VersionLink parameters

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

- Example response

```
{
 "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"
      }
    ]
  }
 ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.1.2 Querying the Version of a Specified VPCEP API

Function

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

URI

GET /{version}

Request

- Parameters

  Table 4-4 Request parameters

Parameter	Mandatory	Type	Description
version	No	String	Specifies the version to be queried. The value starts with v, for example, v1. If this parameter is left blank, versions of all APIs are queried.

- Example request
  GET https://{endpoint}/v1

Response

- Parameters

  Table 4-5 Response parameters

Parameter	Type	Description
version	Object	Lists the versions of VPCEP APIs. For details, see **Table 4-6**.

Table 4-6 VersionModel parameters

Parameter	Type	Description
status	String	Specifies the version status. • **CURRENT**: indicates a major version. • **SUPPORT**: indicates an earlier version which is still supported. • **DEPRECATED**: indicates a deprecated version that may be deleted later.
id	String	Specifies the version ID.
updated	String	Specifies the time when the API version was released. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
version	String	Specifies the supported version.
min_version	String	Specifies the microversion number. If the APIs do not support microversions, the value is left blank.
links	Array of objects	Specifies the API URL. For details, see **Table 4-7**.

Table 4-7 VersionLink parameters

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

- Example response

```
{
 "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"
      }
    ]
  }
 }
}
```

Status Code

For details about status codes, see **Status Code**.

4.2 APIs for Managing 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 endpoint service.

NOTE

This API is an asynchronous interface. If it is successfully invoked, status code **200** is returned, indicating that the request has been successfully delivered. It takes 1 to 2 minutes to create a VPC endpoint service. You can view the creation result by performing operations in **Querying Details of a VPC Endpoint Service**.

URI

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

Table 4-8 describes the required parameters.

Table 4-8 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

Table 4-9 Request parameters

Parameter	Mandatory	Type	Description
port_id	Yes	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. For details, see response field **vip_port_id** in section "Querying Details of a Load Balancer" in the *Elastic Load Balancing API Reference*. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. For details, see section "Querying NICs of an ECS" in the *Elastic Cloud Server API Reference*. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) **NOTE** • To create a VPC endpoint service, the CIDR block of the VPC where the VPC endpoint service is deployed 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.
vip_port_id	No	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound.

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 the format: **regionName.serviceId**. • If you specify this parameter, the VPC endpoint service name is in the format: **regionName.serviceName.serviceId**.
vpc_id	Yes	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs. For details, see response field **id** in section "Querying VPC Details" in the *Virtual Private Cloud API Reference*.
approval_enabled	No	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 **Pending acceptance** state until the owner of the associated VPC endpoint service approves the connection. The default value is **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: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can view those VPC endpoint services that are configured by operations people and are visible and accessible to all users. For detailed steps, see **Querying Public VPC Endpoint Services**. Perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoint services of the gateway type and interface type.

Parameter	Mandatory	Type	Description
server_type	Yes	String	Specifies the resource type. • **VM**: Select this value if the backend resource is an ECS. Backend resources of this type serve as servers. • **VIP**: Select this value if the backend resource is a virtual IP address that functions as a physical server hosting virtual resources. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) • **LB**: Select this value if the backend resource is a load balancer. Backend resources of this type suit services that receive high access traffic and demand high reliability and disaster recovery (DR) performance.
ports	Yes	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-10**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, 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.

Parameter	Mandatory	Type	Description
tcp_proxy	No	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.
tags	No	Array of objects	Lists the resource tags. For details, see **Table 4-11**. A maximum of 10 tags can be added to each VPC endpoint service.

Table 4-10 Port mapping parameters

Parameter	Mandatory	Type	Description
client_port	No	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**
server_port	No	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**
protocol	No	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

Table 4-11 ResourceTags parameters

Parameter	Mandatory	Type	Description
key	No	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	No	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

- Example request

```
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"
  }
 ]
}
```

Response

- Parameter description

Table 4-12 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service.
port_id	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.)
vip_port_id	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound. This parameter is returned only when the backend resource of the VPC endpoint service is a virtual IP address. If the virtual IP address cannot be used for VPCEP, this parameter is not returned.
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. There are two types of VPC endpoint services: interface and gateway. • Gateway: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.
server_type	String	Specifies the resource type. • **VM**: indicates the ECS. • **VIP**: indicates the virtual IP address. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) • **LB**: shared load balancer
vpc_id	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs.
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 **Pending acceptance** state until the owner of the associated VPC endpoint service approves the connection.

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.
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
ports	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-13**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, service ports and terminal ports of all these endpoint services cannot be duplicated when the protocol is the same.

Parameter	Type	Description
tcp_proxy	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-14**.

Table 4-13 Port mapping parameters

Parameter	Type	Description
client_port	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**

Parameter	Type	Description
protocol	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

Table 4-14 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

- Example response

```
{
   "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",
   "service_type":"interface",
   "server_type":"VM",
   "project_id":"6e9dfd51d1124e8d8498dce894923a0d",
   "created_at":"2018-01-30T07:42:01Z",
   "ports":
           [
            {
               "client_port":8080,
               "server_port":90,
               "protocol":"TCP"
            },
            {
               "client_port":8081,
               "server_port":80,
               "protocol":"TCP"
            }
           ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.2.2 Querying Details of a VPC Endpoint Service

Function

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

URI

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

Table 4-15 describes the required parameters.

Table 4-15 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Request

- Parameter description

  None

- Example request

  This request is to query details 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
  ```

Response

- Parameter description

  Table 4-16 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service.

Parameter	Type	Description
port_id	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.)
vip_port_id	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound. This parameter is returned only when the backend resource of the VPC endpoint service is a virtual IP address. If the virtual IP address cannot be used for VPCEP, this parameter is not returned.
service_name	String	Specifies the name of the VPC endpoint service.
server_type	String	Specifies the resource type. • **VM**: indicates the ECS. • **VIP**: indicates the virtual IP address. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) • **LB**: shared load balancer
vpc_id	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs.

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 **Pending acceptance** state until the owner of the associated VPC endpoint service approves the connection.
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: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.

Parameter	Type	Description
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cidr_type	String	Specifies the network segment type. The type can be **public** or **internal**. • **public**: indicates the public subnet CIDR block. • **internal**: indicates the private subnet CIDR block. The default value is **internal**. **NOTE** Only gateway VPC endpoints support parameter **cidr_type**.
ports	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-17**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, service ports and terminal ports of all these endpoint services cannot be duplicated when the protocol is the same.

Parameter	Type	Description
tcp_proxy	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-18**.
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint service changes to **failed**. For details, see **Table 4-19**.

Table 4-17 Port mapping parameters

Parameter	Type	Description
client_port	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**
protocol	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

Table 4-18 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

Table 4-19 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
    "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",
    "service_type":"interface",
    "server_type":"VM",
    "project_id":"6e9dfd51d1124e8d8498dce894923a0d",
```



```
  "created_at":"2018-01-30T07:42:01Z",
  "ports":
          [
           {
              "client_port":8080,
              "server_port":90,
              "protocol":"TCP"
           },
           {
              "client_port":8081,
              "server_port":80,
              "protocol":"TCP"
           }
          ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.2.3 Modifying a VPC Endpoint Service

Function

This API is used to modify a VPC endpoint service.

URI

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

Table 4-20 describes the required parameters.

Table 4-20 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Request

- Parameter description

Table 4-21 Request parameters

Parameter	Mandatory	Type	Description
approval_enabled	No	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 unavailable until the owner of the associated VPC endpoint service approves the connection. The default value is **true**.
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 (-).
ports	No	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-22**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, 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.

Parameter	Mandatory	Type	Description
port_id	No	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. For details, see response field **vip_port_id** in section "Querying Details of a Load Balancer" in the *Elastic Load Balancing API Reference*. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. For details, see section "Querying NICs of an ECS" in the *Elastic Cloud Server API Reference*. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.)
vip_port_id	No	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound.

Parameter	Mandatory	Type	Description
tcp_proxy	No	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.

Table 4-22 Port mapping parameters

Parameter	Mandatory	Type	Description
client_port	No	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**

Parameter	Mandatory	Type	Description
server_port	No	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**
protocol	No	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

- Example request

  This request is to modify the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed88**.

```
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"
          }
        ]
}
```

Response

- Parameter description

Table 4-23 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service.

Parameter	Type	Description
port_id	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.)
vip_port_id	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound.
service_name	String	Specifies the name of the VPC endpoint service.
server_type	String	Specifies the resource type. • **VM**: indicates the ECS. • **VIP**: indicates the virtual IP address. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) • **LB**: shared load balancer
vpc_id	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs.

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 **Pending acceptance** state until the owner of the associated VPC endpoint service approves the connection.
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: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.

Parameter	Type	Description
updated_at	String	Specifies the update time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cidr_type	String	Specifies the network segment type. The type can be **public** or **internal**. • **public**: indicates the public subnet CIDR block. • **internal**: indicates the private subnet CIDR block. The default value is **internal**. **NOTE** Only gateway VPC endpoints support parameter **cidr_type**.
ports	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-24**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, service ports and terminal ports of all these endpoint services cannot be duplicated when the protocol is the same.

Parameter	Type	Description
tcp_proxy	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-25**.

Table 4-24 Port mapping parameters

Parameter	Type	Description
client_port	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**

Parameter	Type	Description
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**
protocol	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

Table 4-25 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

- Example response

```
{
   "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",
   "service_type":"interface",
   "server_type":"VM",
   "project_id":"6e9dfd51d1124e8d8498dce894923a0d",
   "created_at":"2018-01-30T07:42:01Z",
   "ports":
           [
            {
               "client_port":8080,
               "server_port":90,
               "protocol":"TCP"
            },
            {
               "client_port":8081,
               "server_port":80,
               "protocol":"TCP"
            }
           ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.2.4 Deleting a VPC Endpoint Service

Function

This API is used to delete a VPC endpoint service.

URI

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

Table 4-26 describes the required parameters.

Table 4-26 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Request

- Parameter description

  None

- Example request

  This request is to delete the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed88**.

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

Response

None

Status Code

For details about status codes, see **Status Code**.

4.2.5 Querying VPC Endpoint Services

Function

This API is used to query VPC endpoint services.

URI

GET /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}

Table 4-27 describes the required parameters.

Table 4-27 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

Table 4-28 Request 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.
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 is connectable. • **failed**: The VPC endpoint service failed to be created. • **deleting**: The VPC endpoint service is being deleted.

Parameter	Mandatory	Type	Description
sort_key	No	String	Specifies the sorting field of the VPC endpoint service list. The field can be: • **created_at**: VPC endpoint services are sorted by creation time. • **updated_at**: VPC endpoint services are sorted by update time. The default field is **created_at**.
sort_dir	No	String	Specifies the sorting method of the VPC endpoint service list. The method 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**.
limit	No	Integer	Specifies the maximum number of VPC endpoint services displayed on each page. The number ranges from **0** to **1000** and is generally **10**, **20**, or **50**. The default number is **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.

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

Response

- Parameter description

Table 4-29 Response parameters

Parameter	Type	Description
endpoint_services	Array of objects	Lists the VPC endpoint services. For details, see **Table 4-30**.

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

Table 4-30 endpoint_service parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service.
port_id	String	Specifies the ID for identifying the backend resource of the VPC endpoint service. The ID is in the form of the UUID. The values are as follows: • If the backend resource is a load balancer, the value is the ID of the port bound to the private IP address of the load balancer. • If the backend resource is an ECS, the value is the NIC ID of the ECS where the VPC endpoint service is deployed. • If the backend resource is a virtual IP address, the value is the NIC ID of the physical server where virtual resources are created. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.)
vip_port_id	String	Specifies the ID of the virtual NIC to which the virtual IP address is bound. This parameter is returned only when the backend resource of the VPC endpoint service is a virtual IP address. If the virtual IP address cannot be used for VPCEP, this parameter is not returned.
service_name	String	Specifies the name of the VPC endpoint service.
server_type	String	Specifies the resource type. • **VM**: indicates the ECS. • **VIP**: indicates the virtual IP address. (This value will not be returned because virtual IP addresses cannot be used. The LB type is recommended.) • **LB**: shared load balancer

Parameter	Type	Description
vpc_id	String	Specifies the ID of the VPC to which the backend resource of the VPC endpoint service belongs.
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 **Pending acceptance** state until the owner of the associated VPC endpoint service approves the connection.
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: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.

Parameter	Type	Description
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
ports	Array of objects	Lists the port mappings opened to the VPC endpoint service. For details, see **Table 4-31**. Duplicate port mappings are not allowed in the same VPC endpoint service. If multiple VPC endpoint services share the same **port_id** value, service ports and terminal ports of all these endpoint services cannot be duplicated when the protocol is the same.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-32**.
connection_count	Integer	Specifies the number of **Creating** or **Accepted** VPC endpoints under the VPC endpoint service.

Parameter	Type	Description
tcp_proxy	String	Specifies whether the client IP address and port number or **marker_id** information is transmitted to the server. The following methods are supported: • TCP TOA: The client information is inserted into field **tcp option** and transmitted to the server. **NOTE** TCP TOA is supported only when the backend resource is OBS. • Proxy Protocol: The client information is inserted into field **tcp payload** and transmitted to the server. This parameter is available only when the server can parse fields **tcp option** and **tcp payload**. The values are as follows: • **close**: The TOA and Proxy Protocol methods are neither used. • **toa_open**: The TOA method is used. • **proxy_open**: The Proxy Protocol method is used. • **open**: The TOA and Proxy Protocol methods are both used. • **proxy_vni**: The TOA Protocol method is not used. Proxy and virtual network ID are used. The default value is **close**.
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint service changes to **failed**. For details, see **Table 4-33**.

Table 4-31 Port mapping parameters

Parameter	Type	Description
client_port	Integer	Specifies the port for accessing the VPC endpoint. This port is provided by the VPC endpoint, allowing you to access the VPC endpoint service. Supported range: **1** to **65535**

Parameter	Type	Description
server_port	Integer	Specifies the port for accessing the VPC endpoint service. This port is provided by the backend service to provide services. Supported range: **1** to **65535**
protocol	String	Specifies the protocol used in port mappings. The protocol can be **TCP**.

Table 4-32 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

Table 4-33 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
  "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":90,
              "protocol":"TCP"
            },
            {
              "client_port":8081,
```



```
                "server_port":80,
                "protocol":"TCP"
              }
            ],
            "project_id":"6e9dfd51d1124e8d8498dce894923a0d",
            "created_at":"2018-01-30T07:42:01Z",
            "update_at":"2018-01-30T07:42:01Z"
          }
    ],
    "total_count":100
}
```


4.2.6 Querying Connections of a VPC Endpoint Service

Function

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

URI

GET /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}

Table 4-34 describes the required parameters.

Table 4-34 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Table 4-35 Query parameters

Parameter	Mandatory	Type	Description
id	No	String	Specifies the unique ID of the VPC endpoint.
marker_id	No	String	Specifies the packet ID of the VPC endpoint.

Parameter	Mandatory	Type	Description
status	No	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **accepted**: The VPC endpoint has been accepted. • **rejected**: The VPC endpoint has been rejected. • **failed**: The VPC endpoint service failed to be created.
sort_key	No	String	Specifies the sorting field of the VPC endpoint list, which can be: • **created_at**: VPC endpoints are sorted by creation time. • **updated_at**: VPC endpoints are sorted by update time. The default field is **created_at**.
sort_dir	No	String	Specifies the sorting method of the VPC endpoint list, which can be: • **desc**: indicates that VPC endpoints are sorted in the descending order. • **asc**: indicates that VPC endpoints are sorted in ascending order. The default method is **desc**.
limit	No	Integer	Specifies the maximum number of connections displayed on each page. The number ranges from **0** to **1000** and is generally **10**, **20**, or **50**. The default value is **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.

Request

- Parameter description
  None

- Example request

  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
  ```

Response

- Parameter description

Table 4-36 Response parameters

Parameter	Type	Description
connections	Array of objects	Lists the connections. For details, see **Table 4-37**.
total_count	Integer	Specifies the total number of VPC endpoints that meet the search criteria. The number is not affected by the limit or offset.

Table 4-37 Connection parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
created_at	String	Specifies the creation time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
domain_id	String	Specifies the user's domain ID.

Parameter	Type	Description
status	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **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 is being deleted.
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint service changes to **failed**. For details, see **Table 4-38**.

Table 4-38 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
  "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 Code

For details about status codes, see **Status Code**.

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.

URI

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

Table 4-39 describes the required parameters.

Table 4-39 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Request

- Parameter description

  Table 4-40 Request parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies whether to accept or reject a VPC endpoint for a VPC endpoint service. • **receive**: means to accept the VPC endpoint. • **reject**: means to reject the VPC endpoint.
endpoints	Yes	Array of strings	Lists VPC endpoint IDs. Each request accepts or rejects only one VPC endpoint.

- Example request

This request is to accept VPC endpoint **705290f3-0d00-41f2-aedc-71f09844e879** to connect to VPC endpoint service **4189d3c2-8882-4871-a3c2-d380272eed88**.

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

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

Response

- Parameter description

Table 4-41 Response parameters

Parameter	Type	Description
connections	Array of objects	Lists the connections. For details, see **Table 4-42**.

Table 4-42 Connection parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint.
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
created_at	String	Specifies the creation time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
domain_id	String	Specifies the user's domain ID.

Parameter	Type	Description
status	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **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.
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint service changes to **failed**. For details, see **Table 4-43**.

Table 4-43 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
 "connections":
 [
  {
    "id":"4189d3c2-8882-4871-a3c2-d380272eed83",
    "status":"accepted",
    "marker_id":422321321312321321,
    "domain_id":"6e9dfd51d1124e8d8498dce894923a0d",
    "created_at":"2018-01-30T07:42:01Z",
    "updated_at":"2018-01-30T07:42:01Z"
     }
  ]
}
```

or

```
{
 "error_code": "Endpoint.2013"
 "error_msg": "The endpoint does not belong to the endpoint service."
}
```


Status Code

For details about status codes, see **Status Code**.

4.2.8 Querying the Whitelist Records of a VPC Endpoint Service

Function

This API is used to query the whitelist records of a VPC endpoint service.

NOTE

Your ID is in the whitelist of your own VPC endpoint service by default.

URI

GET /v1/{project_id}/vpc-endpoint-services/{vpc_endpoint_service_id}/permissions?permission={permission}&sort_key={sort_key}&sort_dir={sort_dir}&limit={limit}&offset={offset}

Table 4-44 describes the required parameters.

Table 4-44 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Table 4-45 Query parameters

Parameter	Mandatory	Type	Description
permission	No	String	Specifies the authorized ID. The format is the **iam:domain::***domain_id*. *domain_id* indicates the ID, for example, iam:domain::6e9dfd51d1124e8d8498dce894923a0d Fuzzy search is supported.

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the maximum number of whitelist records displayed on each page. The value ranges from **0** to **500** and is generally **10**, **20**, or **50**. The default value is **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.
sort_key	No	String	Specifies the sorting field of the whitelist records. The value is **created_at**, indicating that the whitelist records are sorted by creation time.
sort_dir	No	String	Specifies the sorting method of the whitelist record list. The value can be: ● **desc**: indicates that whitelist records are sorted in the descending order. ● **asc**: indicates that whitelist records are sorted in the ascending order. The default method is **desc**.

Request

- Parameter description

  None

- Example request

  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
  ```

Response

- Parameter description

Table 4-46 Response parameters

Parameter	Type	Description
permissions	Array of objects	Lists the whitelist records. For details, see **Table 4-47**.
total_count	Integer	Specifies the total number of whitelist records that meet the search criteria. The number is not affected by the limit or offset.

Table 4-47 Whitelist record parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the permission.
permission	String	Lists the whitelist records. The permission format is **iam:domain::6e9dfd51d1124e8d8498dce894923a0d** or *. * indicates all users can connect to the VPC endpoint service. **6e9dfd51d1124e8d8498dce894923a0d** indicates the domain ID of the user.
created_at	String	**created_at**: indicates the time of adding the whitelist record. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.

- Example response

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

Status Code

For details about status codes, see **Status Code**.

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

Function

This API is used to batch add to or delete whitelist records from a VPC endpoint service.

NOTE

Your ID is in the whitelist of your own VPC endpoint service by default.

URI

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

Table 4-48 describes the required parameters.

Table 4-48 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_service_id	Yes	Specifies the ID of the VPC endpoint service.

Request

- Parameter description

Table 4-49 Request parameters

Parameter	Mandatory	Type	Description
permissions	Yes	Array of strings	Lists the whitelist records. The record is in the **iam:domain::***domain_id* format. Fields are described as follows: ● **iam:domain::** is fixed. ● *domain_id* indicates the ID of the target user. The value contains 1 to 64 characters, including letters, digits, and asterisks (*). If you enter an asterisk, this VPC endpoint service allows access from any VPC endpoints. Example: **iam:domain::6e9dfd51d1124e8d8498dce894923a0dd**
action	Yes	String	Specifies the operation to be performed. The value is **add** or **remove**.

- Example request

  This request is to add a whitelist record to the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed88**.

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/permissions/action
{
   "permissions":
     [
       "iam:domain::fc973eea581490997e82ea11a1d0101"
     ],
   "action":"add"
}
```

  This request is to delete a whitelist record from the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed88**.

```
POST https://{endpoint}/v1/{project_id}/vpc-endpoint-services/4189d3c2-8882-4871-a3c2-d380272eed88/iam:domain::5fc973eea581490997e82ea11a1d0101/action
{
  "permissions":
    [
      "iam:domain::5fc973eea581490997e82ea11a1d0101"
    ],
   "action":"remove"
}
```

Response

- Parameter description

Table 4-50 Response parameters

Parameter	Type	Description
permissions	Array of strings	Lists the whitelist records. The permission format is **iam:domain::6e9dfd51d1124e8d8498dce894923a0d** or *. * indicates all users can connect to the VPC endpoint service. **6e9dfd51d1124e8d8498dce894923a0d** indicates the domain ID of the user.

- Example response

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

Status Code

For details about status codes, see **Status Code**.

4.3 APIs for Managing VPC Endpoints

4.3.1 Querying Public VPC Endpoint Services

Function

This API is used to query public VPC endpoint services. These services are created by operations people and can be visible to and assessed by all users.

URI

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

Table 4-51 describes the required parameters.

Table 4-51 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

Table 4-52 Request 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 value is **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.
endpoint_service_name	No	String	Specifies the name of the public VPC endpoint service. The value is not case-sensitive and supports fuzzy match.
id	No	String	Specifies the unique ID of the public VPC endpoint service.
sort_key	No	String	Specifies the sorting field of the VPC endpoint service list. The field can be: • **created_at**: VPC endpoint services are sorted by creation time. • **updated_at**: VPC endpoint services are sorted by update time. The default field is **created_at**.

Parameter	Mandatory	Type	Description
sort_dir	No	String	Specifies the sorting method of the VPC endpoint service list. The method 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**.

- Example request
  GET https://{endpoint}/v1/{project_id}/vpc-endpoint-services/public

Response

- Parameter description

Table 4-53 Response parameters

Parameter	Type	Description
endpoint_services	Array of objects	Lists the VPC endpoint services. For details, see **Table 4-54**.
total_count	Integer	Specifies the total number of public VPC endpoint services that meet the search criteria. The number is not affected by the limit or offset.

Table 4-54 endpoint_service parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the public VPC endpoint service.
owner	String	Specifies the owner of the VPC endpoint service.
service_name	String	Specifies the name of the public VPC endpoint service.

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service. The value can be: • Gateway: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
is_charge	Boolean	Specifies whether the associated VPC endpoint carries a charge. • **true**: indicates that the associated VPC endpoint carries a charge. • **false**: indicates that the associated VPC endpoint does not a charge.

- Example response

```
{
 "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
}
```


Status Code

For details about status codes, see **Status Code**.

4.3.2 Querying Basic Information of a VPC Endpoint Service

Function

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

URI

GET /v1/{project_id}/vpc-endpoint-services/describe?endpoint_service_name={endpoint_service_name}&id={endpoint_service_id}

Table 4-55 describes the required parameters.

Table 4-55 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Table 4-56 Query parameters

Parameter	Mandatory	Type	Description
endpoint_service_name	No **NOTE** Either this parameter or the **id** parameter must be selected.	String	Specifies the name of the VPC endpoint service.

Parameter	Mandatory	Type	Description
id	No **NOTE** Either this parameter or the **endpoint_service_name** parameter must be selected.	String	Specifies the unique ID of the VPC endpoint service.

Request

- Parameter description

  None

- Example request

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

Response

- Parameter description

Table 4-57 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint service.
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: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.
created_at	String	Specifies the creation time of the VPC endpoint service. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
is_charge	Boolean	Specifies whether the associated VPC endpoint carries a charge. • **true**: indicates that the associated VPC endpoint carries a charge. • **false**: indicates that the associated VPC endpoint does not a charge.

- Example response

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


Status Code

For details about status codes, see **Status Code**.

4.3.3 Creating a VPC Endpoint

Function

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

URI

POST /v1/{project_id}/vpc-endpoints

Table 4-58 describes the required parameters.

Table 4-58 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

Table 4-59 Request parameters

Parameter	Mandatory	Type	Description
subnet_id	No **NOTE** This parameter is mandatory to create an interface VPC endpoint.	String	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 response field **id** in section "Querying Subnet Details" in the *Virtual Private Cloud API Reference*. This parameter is mandatory only when you create a VPC endpoint for connecting to an interface VPC endpoint service. **NOTE** • The CIDR block of the VPC subnet 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.
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 performing operations in **Querying Basic Information of a VPC Endpoint Service**.
vpc_id	Yes	String	Specifies the ID of the VPC where the VPC endpoint is to be created. For details, see response field **id** in section "Querying VPC Details" in the *Virtual Private Cloud API Reference*.

Parameter	Mandatory	Type	Description
enable_dns	No	Boolean	Specifies whether to create a private domain name. • **true**: A private domain name is created. • **false**: A private domain name is not created. The default value is **false**. **NOTE** When a VPC endpoint for connecting to a gateway VPC endpoint service is created, no private domain name is created no matter **enable_dns** is set to **true** or **false**.
tags	No	Array of objects	Lists the resource tags. For details, see **Table 4-60**. A maximum of 10 tags can be added to each VPC endpoint.
routetables	No	Array of String	Lists the IDs of route tables. For details, see response field **id** in section "Querying a VPC Route" in the *Virtual Private Cloud API Reference*. This parameter is mandatory only when you create a VPC endpoint for connecting to a gateway VPC endpoint service. **NOTE** If this parameter is not configured, use the default route table.
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 only when you create a VPC endpoint for connecting to an interface VPC endpoint service.

Parameter	Mandatory	Type	Description
whitelist	No	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. IPv4 addresses or CIDR blocks can be specified to control access when you create a VPC endpoint. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
enable_whitelist	No	Boolean	Specifies whether access control is enabled.

Table 4-60 ResourceTags parameters

Parameter	Mandatory	Type	Description
key	No	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	No	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

- Example request

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

```
{
"subnet_id": "68bfbcc1-dff2-47e4-a9d4-332b9bc1b8de",
"vpc_id": "84758cf5-9c62-43ae-a778-3dbd8370c0a4",
"tags":[
        {
            "key":"test1",
            "value":"test1"
        }
    ],
"endpoint_service_id":"e0c748b7-d982-47df-ba06-b9c8c7650c1a",
"enable_dns":true
}
```

Response

- Parameter description

Table 4-61 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint.
service_type	String	Specifies the type of the VPC endpoint service that is associated with the VPC endpoint. • Gateway: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can view those VPC endpoint services that are configured by operations people and are visible and accessible to all users. For detailed steps, see **Querying Public VPC Endpoint Services**. Perform the operations in **Creating a VPC Endpoint Service** to create an interface VPC endpoint service.
status	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **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.

Parameter	Type	Description
active_status	String	Specifies the status. • **arrear_frozen**: frozen due to arrears • **verify_frozen**: frozen due to lack of real-name authentification • **police_frozen**: frozen for legal management • **illegal_frozen**: frozen due to abuse • **partner_frozen**: frozen for partnership • **active**: The is normal.
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.
enable_dns	Boolean	Specifies whether to create a private domain name. • **true**: A private domain name is created. • **false**: A private domain name is not created. **NOTE** When a VPC endpoint for connecting to a gateway VPC endpoint service is created, no private domain name is created no matter **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 only available when **enable_dns** is set to **true**.
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in the UUID format.
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created.

Parameter	Type	Description
created_at	String	Specifies the creation time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-62**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist is returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
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, the whitelist is not enabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.

Parameter	Type	Description
routetables	Array of strings	Lists the IDs of route tables. If you do not specify this parameter, the route table ID of the VPC is returned. This parameter is available only when you create a VPC endpoint for connecting to a gateway VPC endpoint service.
enable_status	String	Specifies whether to enable the endpoint. • **enable**: The endpoint will be enabled. • **disable**: The endpoint will be disabled.

Table 4-62 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

- Example response

```
{
   "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
   "status": "creating",
   "service_type": "interface",
   "marker_id": 322312312312,
   "active_status":"active",
   "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,
   "enable_status":"enable",
   "created_at": "2018-01-30T07:42:01Z",
   "update_at": "2018-01-30T07:42:01Z",
   "tags": [
      {
         "key": "test1",
```



```
        "value": "test1"
      }
    ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.3.4 Querying Details of a VPC Endpoint

Function

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

URI

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

Table 4-63 describes the required parameters.

Table 4-63 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_id	Yes	Specifies the ID of the VPC endpoint.

Request

- Parameter description

  None

- Example request

  This request is to query details of the VPC endpoint service whose ID is **4189d3c2-8882-4871-a3c2-d380272eed83**.

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

Response

- Parameter description

Table 4-64 Response parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint.
service_type	String	Specifies the type of the VPC endpoint service that is associated with the VPC endpoint. • Gateway: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them.
status	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **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
active_status	String	Specifies the status. • **arrear_frozen**: frozen due to arrears • **verify_frozen**: frozen due to lack of real-name authentification • **police_frozen**: frozen for legal management • **illegal_frozen**: frozen due to abuse • **partner_frozen**: frozen for partnership • **active**: The is normal.
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.
enable_dns	Boolean	Specifies whether to create a private domain name. • **true**: A private domain name is created. • **false**: A private domain name is not created. **NOTE** When a VPC endpoint for connecting to a gateway VPC endpoint service is created, no private domain name is created no matter **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 only available when **enable_dns** is set to **true**.

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. • The connection approval function is enabled for the VPC endpoint service, and the connection has been approved. The status of the VPC endpoint can be **Accepted** or **Rejected**. The **Rejected** status only appears when the VPC endpoint is accepted and then rejected.
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created.
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in the UUID format. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
created_at	String	Specifies the creation time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-65**.

Parameter	Type	Description
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint changes to **failed**. For details, see **Table 4-66**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist is returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
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, the whitelist is not enabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Lists the IDs of route tables. If you do not specify this parameter, the route table ID of the VPC is returned. This parameter is available only when you create a VPC endpoint for connecting to a gateway VPC endpoint service.
enable_status	String	Specifies whether to enable the endpoint. • **enable**: The endpoint will be enabled. • **disable**: The endpoint will be disabled.

Table 4-65 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

Table 4-66 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
   "id": "4189d3c2-8882-4871-a3c2-d380272eed83",
   "status": "accepted",
   "service_type": "interface",
   "marker_id": 322312312312,
   "active_status":"active",
   "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": "2018-01-30T07:42:01Z",
   "update_at": "2018-01-30T07:42:01Z",
   "tags": [
      {
         "key": "test1",
         "value": "test1"
      }
   ]
}
```

Status Code

For details about status codes, see **Status Code**.

4.3.5 Deleting a VPC Endpoint

Function

This API is used to delete a VPC endpoint.

URI

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

For detailed about the parameters, see **Table 4-67**.

Table 4-67 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
vpc_endpoint_id	Yes	Specifies the ID of the VPC endpoint.

Request

- Parameter description

  None

- Example request

  This request is to delete the VPC endpoint whose ID is **4189d3c2-8882-4871-a3c2-d380272eed83**.

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

Response

None

Status Code

For details about status codes, see **Status Code**.

4.3.6 Querying VPC Endpoints

Function

This API is used to query VPC endpoints.

URI

GET /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}

Table 4-68 describes the required parameters.

Table 4-68 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

Table 4-69 Request 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.
vpc_id	No	String	Specifies the ID of the VPC where the VPC endpoint is to be created.
id	No	String	Specifies the unique ID of the VPC endpoint.
limit	No	Integer	Specifies the maximum number of VPC endpoints displayed on each page. The number ranges from **0** to **1000** and is generally **10**, **20**, or **50**. The default number is **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.

Parameter	Mandatory	Type	Description
sort_key	No	String	Specifies the sorting field of the VPC endpoint list. The field can be: • **created_at**: VPC endpoints are sorted by creation time. • **updated_at**: VPC endpoints are sorted by update time. The default field is **created_at**.
sort_dir	No	String	Specifies the sorting method of the VPC endpoint list. The method can be: • **desc**: VPC endpoints are sorted in descending order. • **asc**: VPC endpoints are sorted in ascending order. The default method is **desc**.

- Example request
  GET https://{endpoint}/v1/{project_id}/vpc-endpoints

Response

- Parameter description

Table 4-70 Response parameters

Parameter	Type	Description
endpoints	Array of objects	Lists the VPC endpoints. For details, see **Table 4-71**.
total_count	Integer	Specifies the total number of VPC endpoints that meet the search criteria. The number is not affected by the limit or offset.

Table 4-71 VPC endpoint parameters

Parameter	Type	Description
id	String	Specifies the unique ID of the VPC endpoint.

Parameter	Type	Description
service_type	String	Specifies the type of the VPC endpoint service that is associated with the VPC endpoint. • Gateway: VPC endpoint services of this type are configured by operations people. You can use them directly without the need to create one by yourselves. • Interface: VPC endpoint services of this type include cloud services configured by operations people and private services created by yourselves. You cannot configure these cloud services, but can use them. You can perform the operations in **Creating a VPC Endpoint** to create VPC endpoints for accessing VPC endpoints of the gateway and interface types.
status	String	Specifies the connection status of the VPC endpoint. • **pendingAcceptance**: The VPC endpoint is pending acceptance. • **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	String	Specifies the status. • **arrear_frozen**: frozen due to arrears • **verify_frozen**: frozen due to lack of real-name authentification • **police_frozen**: frozen for legal management • **illegal_frozen**: frozen due to abuse • **partner_frozen**: frozen for partnership • **active**: The is normal.
endpoint_service_name	String	Specifies the name of the VPC endpoint service.

Parameter	Type	Description
marker_id	Integer	Specifies the packet ID of the VPC endpoint.
endpoint_service_id	String	Specifies the ID of the VPC endpoint service.
enable_dns	Boolean	Specifies whether to create a private domain name. • **true**: A private domain name is created. • **false**: A private domain name is not created. **NOTE** When a VPC endpoint for connecting to a gateway VPC endpoint service is created, no private domain name is created no matter **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 only available when **enable_dns** is set to **true**.
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. • The connection approval function is enabled for the VPC endpoint service, and the connection has been approved. The status of the VPC endpoint can be **Accepted** or **Rejected**. The **Rejected** status only appears when the VPC endpoint is accepted and then rejected.
vpc_id	String	Specifies the ID of the VPC where the VPC endpoint is to be created.
subnet_id	String	Specifies the ID of the subnet in the VPC specified by **vpc_id**. The ID is in the UUID format.
created_at	String	Specifies the creation time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.
updated_at	String	Specifies the update time of the VPC endpoint. The UTC time format is used: YYYY-MM-DDTHH:MM:SSZ.

Parameter	Type	Description
project_id	String	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
tags	Array of objects	Lists the resource tags. For details, see **Table 4-72**.
error	Array of objects	Specifies the error message. This field is returned when the status of the VPC endpoint changes to **failed**. For details, see **Table 4-73**.
whitelist	Array of strings	Specifies the whitelist for controlling access to the VPC endpoint. If you do not specify this parameter, an empty whitelist is returned. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
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, the whitelist is not enabled. This parameter is available only when you create a VPC endpoint for connecting to an interface VPC endpoint service.
routetables	Array of strings	Lists the IDs of route tables. If you do not specify this parameter, the route table ID of the VPC is returned. This parameter is available only when you create a VPC endpoint for connecting to a gateway VPC endpoint service.
enable_status	String	Specifies whether to enable the endpoint. • **enable**: The endpoint will be enabled. • **disable**: The endpoint will be disabled.

Table 4-72 ResourceTags parameters

Parameter	Type	Description
key	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. This parameter cannot be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.
value	String	Specifies the tag value. A tag value contains a maximum of 43 Unicode characters and can be left blank. It cannot contain characters =*<>\,|/ nor start or end with a space.

Table 4-73 Error parameters

Parameter	Type	Description
error_code	String	Specifies the error code.
error_message	String	Specifies the error message.

- Example response

```
{
"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_name":"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",
     "whitelist":["127.0.0.1"],
     "enable_whitelist":true,
     "tags":
       [
         {
           "key":"test1",
           "value":"test1"
         }
       ]
   },
   {
     "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_name":"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",
    "whitelist":["127.0.0.1"],
    "enable_whitelist":true,
    "tags":
      [
        {
          "key":"test1",
          "value":"test1"
        }
      ]
   }
 ],
 "total_count":17
}
```


Status Code

For details about status codes, see **Status Code**.

4.4 Resource Quota

4.4.1 Querying the Quota

Function

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

URI

GET /v1/{project_id}/quotas?type={resource_type}

Table 4-74 describes the required parameters.

Table 4-74 Parameters

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

- Parameter description

  Table 4-75 Request parameters

Parameter	Mandatory	Type	Description
type	No	String	Specifies the resource type. • **endpoint_service**: indicates the VPC endpoint service. • **endpoint**: indicates the VPC endpoint.

- Example request

  This request is to query the quota of VPC endpoint services.

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

Response

- Parameter description

  Table 4-76 Response parameters

Parameter	Type	Description
quotas	Object	Specifies quota details. For details, see **Table 4-77**.

  Table 4-77 Quotas description

Parameter	Type	Description
resources	Array of objects	Lists the resources. For details, see **Table 4-78**.

  Table 4-78 Resource parameters

Parameter	Type	Description
type	String	Specifies the resource type. You can query the quota of resources of a specified type by configuring this parameter. • **endpoint_service**: indicates the VPC endpoint service. • **endpoint**: indicates the VPC endpoint.

Parameter	Type	Description
used	Integer	Specifies the number of created resources. The value ranges from **0** to the value of **quota**.
quota	Integer	Specifies the maximum quota of resources. The value ranges from the default quota value to the maximum quota value.

- Example response

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

Status Code

For details about status codes, see **Status Code**.

4.5 Tag Function

4.5.1 Querying Resources by Tag

Function

This API is used to query resources under the tenant using tags.

URI

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

Table 4-79 describes the required parameters.

Table 4-79 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_type	Yes	String	Specifies the resource type. The type is **endpoint_service** or **endpoint**.

Request

- Parameter description

Table 4-80 Request parameters

Parameter	Mandatory	Type	Description
tags	No	List<tag>	Lists tags that are included. Each resource to be queried contains a maximum of 10 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Each tag key and each value of the same tag key must be unique. The response returns resources containing all tags in this list. Keys in this list are in an AND relationship while values in each key-value structure are in an OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Type	Description
tags_any	No	List<tag>	Lists any tags that are included. Each resource to be queried contains a maximum of 10 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Each tag key and each value of the same tag key must be unique. The response returns resources containing the tags in this list. Keys in this list are in an OR relationship and values in each key-value structure are also in an OR relationship. If no tag filtering condition is specified, full data is returned.
not_tags	No	List<tag>	Lists tags that are not included. Each resource to be queried contains a maximum of 10 keys. Each tag key can have a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Each tag key and each value of the same tag key must be unique. The response returns resources containing no tags in this list. Keys in this list are in an AND relationship while values in each key-value structure are in an OR relationship. If no tag filtering condition is specified, full data is returned.

Parameter	Mandatory	Type	Description
not_tags_any	No	List<tag>	Lists any tags that are not included. Each resource to be queried contains a maximum of 10 keys. Each tag key has a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Each tag key and each value of the same tag key must be unique. The response returns resources containing no tags in this list. Keys in this list are in an AND relationship while values in each key-value structure are in an OR relationship. If no tag filtering condition is specified, full data is returned.
limit	No	String	Sets the page size. This parameter is unavailable when **action** is set to **count**. The default size is **1000** when **action** is set to **filter**. The maximum value is **1000**, and the minimum size is **1**. The size cannot be a negative number.
offset	No	String	Specifies the index position. This parameter is unavailable when **action** is set to **count**. If **offset** is set to *N*, the resource query starts from the N+1 piece of data. If **action** is set to **filter**, **offset** is **0** by default, indicating that the query starts from the first piece of data. **offset** must be a positive number.
action	Yes	String	Specifies the operation to perform. The operation can only be **filter** (filtering) or **count** (querying the total number). If **action** is set to **filter**, the query is performed based on the filter conditions. If **action** is set to **count**, only the total number of records is returned.

Parameter	Mandatory	Type	Description
matches	No	List<match>	Specifies the search field. The tag key is the field to be matched, for example, **resource_name**. **value** indicates the matched value. The key is a fixed dictionary value and cannot contain duplicate keys or unsupported keys. Check whether fuzzy match is required based on the **key** value. For example, if **key** is set to **resource_name**, fuzzy search (case-insensitive) is performed by default. If **value** is empty, exact match is performed. Most services do not have resources without names. In this case, an empty list is returned. If **key** is **resource_id**, exact match is used. Only **resource_name** for **key** is supported. Other **key** values will be available later.

Table 4-81 Description of field **tag**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. Each tag key contains a maximum of 127 unicode characters but cannot be left blank. The system does not verify the character set of **key** when searching for resources. **key** cannot be empty, an empty string, or spaces. Before using **key**, delete single-byte character (SBC) spaces before and after the value.

Parameter	Mandatory	Type	Description
values	Yes	List<String>	Lists tag values. Each value contains a maximum of 255 Unicode characters. Before using **values**, delete SBC spaces before and after the value. The value can be an empty array but cannot be left blank. If the values are null, it indicates **any_value** (querying any value). The 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.

Table 4-82 Description of field **match**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. Only **resource_name** for **key** is supported. Other **key** values will be available later.
value	Yes	String	Specifies the tag value. Each value contains a maximum of 255 Unicode characters. The character set of **value** is not verified.

- Example request

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

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

  or POST https://{endpoint}/v1/{project_id}/{resource_type}/resource_instances/action

  – Request body 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"
      ]
    }
  ]
}
```

– Request body 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"
        }
    ]
}
```

Response

- Parameter description

Table 4-83 Parameter description

Parameter	Type	Description
resources	List<resource>	N/A
total_count	Integer	Specifies the total number of records.

Table 4-84 Data structure of field **resource**

Parameter	Type	Description
resource_id	String	Specifies the resource ID, which can be **Endpoint Service ID** or **Endpoint ID**.
tags	List<resource_tag>	Lists the tags. If no tag is matched, an empty array is returned.
resource_name	String	Specifies the resource name. If the resource does not have a name, the ID is returned.

Table 4-85 Data structure of field **resource_tag**

Parameter	Type	Description
key	String	Specifies the tag key.

Parameter	Type	Description
value	String	Specifies the tag value.

- Example response
  - Response body when **action** is set to **filter**

```
{
   "resources": [
      {
         "resource_detail": null,
         "resource_id": "cdfs_cefs_wesas_12_dsad",
         "resource_name": "resource1",
         "tags": [
            {
               "key": "key1","value": "value1"
            },
            {
               "key": "key2","value": "value1"
            }
         ]
      }
   ],
   "total_count": 1000
}
```

  - Response body when **action** is set to **count**

```
{
   "total_count": 1000
}
```

Status Code

For details about status codes, see **Status Code**.

4.5.2 Batch Adding or Deleting Tags of a Specified Resource

Function

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

- You can add a maximum of 10 tags to a resource.

URI

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

Table 4-86 describes the required parameters.

Table 4-86 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Specifies the resource type. The type is **endpoint_service** or **endpoint**.
resource_id	Yes	String	Specifies the resource ID, which can be **Endpoint Service ID** or **Endpoint ID**.

Request

- Parameter description

Table 4-87 Request parameters

Parameter	Mandatory	Type	Description
tags	No	List<resource_tag>	Lists the tags. This parameter is mandatory for common tenants.
action	Yes	String	Specifies the operation to be performed, which can be **create** or **delete**

Table 4-88 Data structure of field **resource_tag**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. A tag key contains a maximum of 36 Unicode characters. The key meets the requirements in **Tag Character Set Specifications**.

Parameter	Mandatory	Type	Description
value	This parameter is mandatory when **action** is set to **create** and optional when **action** is set to **delete**.)	String	Specifies the tag value. Each value contains a maximum of 43 Unicode characters. If **value** is specified, tags are deleted by key and value. If **value** is not specified, tags are deleted by key. The value meets the requirements in **Tag Character Set Specifications**.

- Example request

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

  or https://{endpoint}/v1/{project_id}/endpoint/{resource_id}/tags/action

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

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

  or

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

Response

None

Status Code

For details about status codes, see **Status Code**.

4.5.3 Querying Tags of a Tenant's Resource

Function

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

URI

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

Table 4-89 describes the required parameters.

Table 4-89 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_type	Yes	String	Specifies the resource type. The value is **endpoint_service** or **endpoint**.

Request

- Parameter description

  None

- Example request

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

  or https://{endpoint}/v1/{project_id}/endpoint/tags GET /v1/{project_id}/{resource_type}/tags

Response

- Parameter description

  Table 4-90 Response parameters

Parameter	Mandatory	Type	Description
tags	Yes	List<resource_tag>	Lists the tags.

Table 4-91 Data structure of field **resource_tag**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key.
values	Yes	List<String>	Lists the tag values.

- Example response

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

Status Code

For details about status codes, see **Status Code**.

4.5.4 Tag Character Set Specifications

Character Set Specifications for Keys

The key should be Unicode characters and cannot be left blank. It cannot contain:

Non-printable ASCII characters (0-31), "=", "*", "<", ">", "\", ",", "|", "/"

Character Set Specifications for Values

The value should be Unicode characters. It cannot contain:

Non-printable ASCII characters (0-31), "=", "*", "<", ">", "\", ",", "|", "/"

5 Application Examples

5.1 Configuring a VPC Endpoint for Communication Across VPCs

Scenarios

VPCEP 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's 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 of the interface type.
- **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, ECSs in the VPC where the VPC endpoint is located can access VPC endpoint service **4189d3c2-8882-4871-a3c2-d380272eed83** using this endpoint.

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

Function

If an error occurs during API calling, a customized error message will be returned. This section describes the meaning of each status code returned by VPCEP.

Error Code Structure Format

```
STATUS CODE 400
{
    "error_code": "EndPoint.0002",
    "error_msg": "Parameter error."
}
```

Error Code Descriptions

NOTE

An error code returned by an API does not correspond to an error message.

Table 6-3 Error codes

Status Code	Error Code	Error Message	Description	Solution
500	EndPoint.0001	System error. Please retry.	System error. Please retry.	Try again later. If the fault persists, contact technical support.
400	EndPoint.0002	Parameter error.	Parameter error.	Check whether the parameter is correct.
401	EndPoint.0003	Authentication failed or authentication information is invalid.	Authentication failed or authentication information is invalid.	Check whether the permission is 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 permission is enabled.
404	EndPoint.0005	The requested resource does not exist.	The requested resource is unavailable.	Check whether input parameters are correct.
400	EndPoint.0006	Invalid limit.	Invalid limit.	Enter a correct limit.
400	EndPoint.0007	Invalid action.	Invalid action.	Enter a correct action.
400	EndPoint.0009	The remote address does not match.	The remote address does not match.	Check whether you have the access permission.
400	EndPoint.0010	Invalid offset.	Invalid offset.	Enter a correct offset.
504	EndPoint.0011	The request body is null.	The request body is null.	Enter the request body again.

Status Code	Error Code	Error Message	Description	Solution
504	EndPoint.0012	The request header is null.	The request header is null.	Enter a request header.
504	EndPoint.0013	The request timed out.	The request timed out.	Try again later. If the fault persists, contact technical support.
400	EndPoint.0014	Invalid project ID.	Invalid project ID.	Enter a correct project ID.
400	EndPoint.0015	Invalid specification.	Invalid specifications.	Enter correct 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 correct sort key.
400	EndPoint.0018	Invalid sort_dir.	Invalid sort DIR.	Enter a correct sort DIR.
400	EndPoint.0019	Invalid status.	Invalid status.	Enter a correct status.
400	EndPoint.0020	Invalid VPC ID.	Invalid VPC ID.	Enter a correct VPC ID.
400	EndPoint.0021	Invalid marker_id.	Invalid marker ID.	Enter a correct marker ID.
400	EndPoint.0022	The number of requests exceeded the limit. Please try later.	The number of requests exceeded the limit. Please try later.	Try again later.
400	EndPoint.0023	Invalid subnet_id.	Invalid subnet ID.	Enter a correct subnet ID.
400	EndPoint.1003	Invalid service name.	Invalid service name.	Enter a correct service name.
400	EndPoint.1004	Invalid request.	Invalid request.	Enter a correct request body.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.1008	Failed to obtain the token.	Failed to obtain the token.	Contact technical support.
400	Endpoint.1018	Quota exceeded.	Quota exceeded.	Contact technical support.
400	EndPoint.1019	Invalid route table ID.	Invalid route table ID.	Enter a correct route table ID.
400	EndPoint.2001	The VPC does not exist.	The VPC does not exist.	Enter a correct VPC ID for the current tenant.
400	EndPoint.2002	The request input parameter is empty.	The request input parameter is empty.	Please input a correct parameter.
400	EndPoint.2003	The endpoint service does not exist.	The VPC endpoint service does not exist.	Enter a 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.
404	EndPoint.2006	The requested endpoint does not exist.	The requested VPC endpoint does not exist.	Enter a correct VPC endpoint.
404	EndPoint.2007	The endpoint information does not exist.	The VPC endpoint information is unavailable.	Enter a correct VPC endpoint and check whether the endpoint is deleted.
404	EndPoint.2008	The endpoint has been deleted.	The VPC endpoint has been deleted.	Check whether the VPC endpoint is deleted.
400	EndPoint.2009	The specification information does not exist.	The specification is unavailable.	Enter correct specifications.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2010	The input parameter subnet ID is empty.	The subnet ID is left blank.	Enter a valid subnet ID.
400	EndPoint.2011	The input parameter VPC ID is empty.	The 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 permission.
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 correct.
400	EndPoint.2014	The endpoint has been 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 to confirm the freezing reason.
400	EndPoint.2016	The endpoint pool IP address does not exist.	The VPC endpoint pool IP address does not exist.	Contact technical support.
400	EndPoint.2017	Invalid endpoint ID.	The ID of the VPC endpoint is invalid.	Enter a correct 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.

Status Code	Error Code	Error Message	Description	Solution
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 a VPC endpoint.	Contact technical support.
400	EndPoint.2023	CIDR is not found.	CIDR is not found.	Contact technical support.
400	EndPoint.2024	shadowVpc or shadowPort does not exist.	**shadowVpc** or **shadowPort** is null.	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 empty.	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 empty.	Enter a valid port number or protocol.
400	EndPoint.2029	The requested endpoint service ID is empty.	The ID of the requested VPC endpoint service is empty.	Enter a valid VPC endpoint service ID.
404	EndPoint.2030	markerId is empty.	**markerId** is empty.	Enter a valid marker ID.
400	EndPoint.2031	Only one endpoint is allowed.	Only one VPC endpoint is allowed.	A VPC endpoint service allows accepting or rejecting only one VPC endpoint.
400	EndPoint.2033	The entered parameter enable_dns is invalid.	Invalid parameter **enable_dns**.	Enter a valid parameter.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2034	The entered parameter enable_dns is incorrect.	The entered parameter **enable_dns** is incorrect.	Enter a valid parameter.
400	EndPoint.2035	The system parameter dns.enable is invalid.	System parameter **dns.enable** is invalid.	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 correct.
400	EndPoint.2038	The pool does not exist.	The resource pool is not found.	Contact technical support.
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 correct IP address.
400	EndPoint.2042	The Port IP is in use.	The IP address of the NIC is in use.	Enter a correct 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 correct IP address.
400	EndPoint.2044	Invalid whitelist.	Invalid whitelist.	Enter valid request parameters.
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.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.2046	Can not modify a specified mac endpoint.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 valid request parameters.
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 endpoint has been disabled.	Contact technical support.
400	EndPoint.2051	The current endpoint is switching to new pool.	The 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 a port.	Contact technical support.
400	EndPoint.3002	Invalid permission.	Invalid permission.	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.

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

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3016	endpointService gateway vlan can't have ip.	The request for accessing the VLAN VPC endpoint service cannot contain IP address information.	Enter a correct request body.
400	EndPoint.3017	Invalid CIDRs.	Invalid CIDRs.	Enter correct CIDRs.
400	EndPoint.3018	endpointService gateway vlan can't have vpcId.	The request for accessing the VLAN VPC endpoint service cannot contain VPC ID information.	Enter a correct request body.
400	EndPoint.3021	Invalid serverType.	Invalid parameter **serverType**.	Enter a valid parameter.
400	EndPoint.3022	Failed to create a network.	Failed to create a network.	Contact technical support.
400	EndPoint.3023	Failed to create a subnet.	Failed to create a subnet.	Contact technical support.
400	EndPoint.3035	Invalid action.	Invalid action.	Enter a correct action.
400	EndPoint.3036	Invalid permissions.	The permission list cannot be empty.	Enter a correct request body.
400	EndPoint.3040	Failed to add a rollback task.	Failed to add a 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 correct request body.
400	EndPoint.3043	The service port is invalid.	Invalid service port.	Enter a correct request body.

Status Code	Error Code	Error Message	Description	Solution
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 correct request body.
400	EndPoint.3045	Other properties cannot be modified in the current endpoint service state.	Modifying other properties in the current endpoint service state is not supported.	Enter a correct request body.
400	EndPoint.3046	The IP address conflicted with an existing endpoint service.	The IP address conflicted with an existing VPC endpoint service.	Enter a correct request body.
400	EndPoint.3048	Invalid netType.	Invalid **netType**.	Enter a valid value.
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 asterisk (*).
400	EndPoint.3051	Endpoint service vip port id is invalid.	Invalid parameter **vip_port_id**.	Enter a correct value.
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 correct 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 correct request body.
400	EndPoint.3054	portId or vipPortId cannot be modified.	**portId** or **vipPortId** cannot be modified.	Enter a correct request body.
400	EndPoint.3055	ip cannot be modified.	**ip** cannot be modified.	Enter a correct 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 of VPC endpoint services supported by a backend resource has been reached.	Contact technical support.
400	EndPoint.3057	cidr cannot be modified.	CIDR cannot be modified.	Enter a correct request body.
400	EndPoint.3058	The domain name is invalid.	Invalid domain name.	Enter a correct domain 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 correct parameter.
400	EndPoint.3063	Invalid port ID.	Invalid port ID.	Enter a correct port ID.
400	EndPoint.3066	The tag cannot be empty.	The tag cannot be empty.	Enter a correct request body.
400	EndPoint.3067	The tag key cannot be duplicated.	The tag key cannot be duplicated.	Enter a correct request body.
400	EndPoint.3068	Tag keys and values should meet relevant requirements.	Tag keys and values must meet relevant requirements.	Enter a correct request body.
400	EndPoint.3069	The maximum number of tags has been reached.	The maximum number of tags has been reached.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.3070	Invalid resource type.	Incorrect resource type.	Contact technical support.
400	EndPoint.3071	The tag value cannot be duplicated.	Tag values cannot be duplicated.	Contact technical support.
400	EndPoint.3072	The tag key size is invalid.	The tag key size is invalid.	Enter a correct tag key.
400	EndPoint.3073	The tag value size is invalid.	The tag value size is invalid.	Enter a correct tag value.
400	EndPoint.3074	The maximum of ports has been reached.	The maximum 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 a subnet.	Contact technical support.
400	EndPoint.4003	Failed to delete the subnet.	Failed to delete the subnet.	Contact technical support.
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.
400	EndPoint.4005	Failed to query the network.	Failed to query the network.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	EndPoint.4006	Failed to create a network.	Failed to create a network.	Contact technical support.
400	EndPoint.4007	Failed to delete the network.	Failed to delete the network.	Contact technical support.
404	EndPoint.4008	Network is unavailable.	Network is unavailable.	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 a port.	Contact technical support.
400	EndPoint.4011	Failed to delete the port.	Failed to delete the port.	Contact technical support.
404	EndPoint.4012	The port is not found.	The port is not found.	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 an 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.

Status Code	Error Code	Error Message	Description	Solution
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.
404	EndPoint.4021	Neutron L3 Agent is not found.	Neutron L3 Agent is not found.	Contact technical support.
400	EndPoint.4025	The specification is being used.	The specification is 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.
404	EndPoint.4030	The route table is not found.	The route table is not found.	Contact technical support.

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 the project ID by calling the IAM API used to query project information based on the 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": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

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 **My Credentials** page, view the project ID (value in the **Project ID** column).

A Change History

Released On	Description
2022-04-12	This issue is the first official release.

