Direct Connect

API Reference

Issue 01

Date 2024-07-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

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

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

Contents

1 Before You Start

1.1 Introduction

Welcome to Direct Connect API Reference. Direct Connect allows you to connect your on-premises data center to the cloud over a dedicated network connection that features stable performance, high transmission speed, low network latency, and secure data transmission. Direct Connect allows you to maximize legacy IT facilities and leverage cloud services to build a flexible, scalable hybrid cloud computing environment.

This document describes how to use application programming interfaces (APIs) to perform operations on Direct Connect resources, such as creating, querying, modifying, or deleting connections. For details about all supported operations, see **API Overview**.

Before you access Direct Connect by calling APIs, get yourself familiar with product concepts. For details, see **Service Overview**.

1.2 API Calling

Direct Connect supports Representational State Transfer (REST) APIs that can be called using HTTPS. For details, see **Calling APIs**.

1.3 Endpoints

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

1.4 Constraints

The number of Direct Connect resources that you can create is determined by your quotas. To view or increase the quotas, see **Notes and Limitations** in the *Direct Connect User Guide*.

For more constraints, see API descriptions.

1.5 Concepts

- Account

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

- User

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

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

- Region

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

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

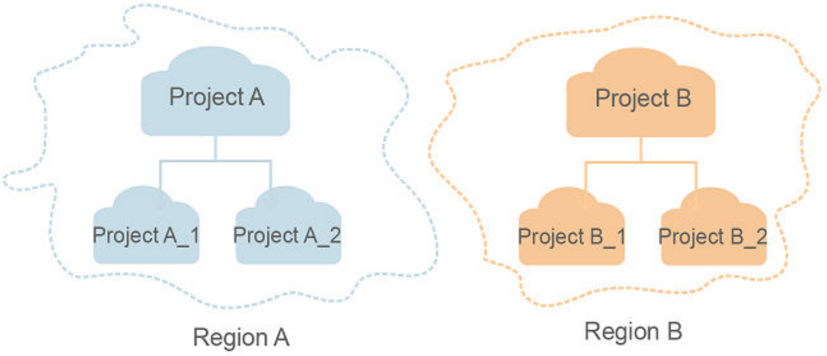
- AZ

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

- Project

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

Figure 1-1 Project isolation model



- Enterprise project

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

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

2 API Overview

Direct Connect provides Huawei-developed REST APIs.

These APIs allow you to use all Direct Connect functions, for example, performing operations on connections, virtual gateways, virtual interfaces, and tags.

Table 2-1 describes the APIs provided by Direct Connect.

Table 2-1 Direct Connect APIs

Resource	APIs
Connections	APIs for querying connection details, updating a connection, deleting a connection, querying the connection list, querying the hosted connection list, creating a hosted connection, querying hosted connection details, updating hosted connection information, and deleting a hosted connection
Virtual gateways	APIs for querying virtual gateway details, updating a virtual gateway, deleting a virtual gateway, querying the virtual gateway list, and creating a virtual gateway
Virtual interfaces	APIs for querying virtual interface details, updating a virtual interface, deleting a virtual interface, querying the virtual interface list, and creating a virtual interface
Tags	APIs for querying project tags, querying resource tags, adding a resource tag, batch adding or deleting resource tags, deleting a resource tag, and querying resources by tag
Quotas	API for querying resource quotas

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

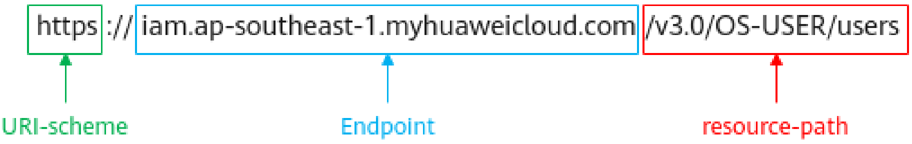
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of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

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

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

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

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

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

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

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

Request Header

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

Common request header fields are as follows.

Table 3-3 Common request header fields

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

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

NOTE

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

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

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date,
Signature=****************
```

(Optional) Request Body

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

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

The following shows an example request (a request body included) of the API for **creating an IAM user**. You can learn about request parameters and related description from this example. The bold parameters need to be replaced for a real request.

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

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date,
Signature=****************

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

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

3.2 Authentication

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

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

AK/SK Authentication

NOTE

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

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

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

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

NOTE

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

Token Authentication

NOTE

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

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

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

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

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

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

3.3 Response

Status Code

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

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

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

Response Header

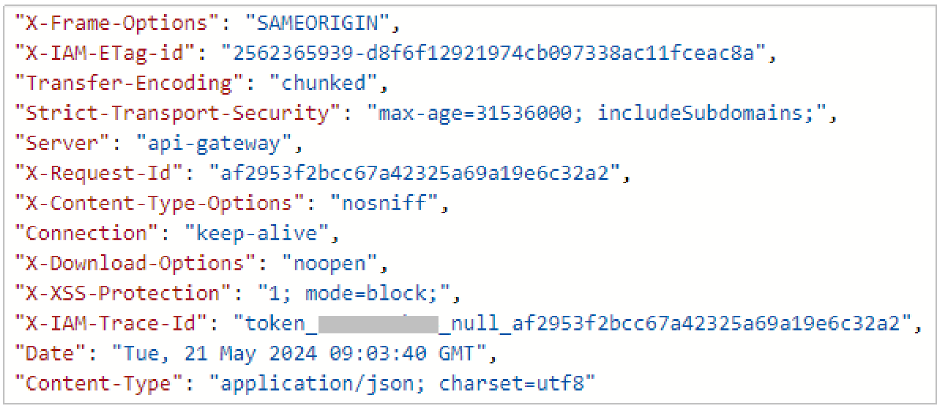
Similar to a request, a response also has a header, for example, **Content-Type**.

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

NOTE

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

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



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

(Optional) Response Body

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

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

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

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

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

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

4 API Usage

Direct Connect has independent endpoints. When API calls are made to Direct Connect, the Direct Connect APIs are used. Therefore, you need pay special attention to the service endpoints used when you make calls to the Direct Connect APIs.

5 API

5.1 Connection

5.1.1 Querying Details About a Connection

Function

This API is used to query details about a connection.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}

Table 5-1 Path Parameters

Parameter	Mandatory	Type	Description
direct_connect_id	Yes	String	Specifies the connection ID. Minimum: **36** Maximum: **36**
project_id	Yes	String	Specifies the project ID.

Table 5-2 Query Parameters

Parameter	Mandatory	Type	Description
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**

Request Parameters

Table 5-3 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-4 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
direct_connect	**DirectConnect** object	Specifies parameters for creating a connection.

Table 5-5 DirectConnect

Parameter	Type	Description
id	String	Specifies the connection ID.
tenant_id	String	Specifies the ID of the project that the instance belongs to.

Parameter	Type	Description
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **100000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
device_id	String	Specifies the ID of the device connected to the connection. Minimum: **0** Maximum: **36**

Parameter	Type	Description
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
charge_mode	String	Specifies the billing mode, which can be **prepayment**, **bandwidth**, or **traffic**. Enumeration values: ● **prepayment** ● **bandwidth** ● **traffic**
provider	String	Specifies the line carrier of a connection. It can be China Telecom, China Unicom, China Mobile, and other carriers that provide the lines.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the connection status. The options are as follows: **ACTIVE**: The connection is in the normal state. **DOWN**: The port for the connection is in the down state, which may cause line faults. **BUILD**: Operations related to the connection are being performed. **ERROR**: The connection configuration is incorrect. Contact customer service to rectify the fault. **PENDING_DELETE**: The connection is being deleted. **DELETED**: The connection has been deleted. **APPLY**: A request for a connection is submitted. **DENY**: A site survey is rejected because the customer fails to meet the requirements. **PENDING_PAY**: The order for the connection is to be paid. **PAID**: The order for the connection has been paid. **PENDING_SURVEY**: A site survey is required for the connection. **LEASED_LINE_DELIVERY**: The carrier is delivering the leased line. Enumeration values: ● **BUILD** ● **PAID** ● **APPLY** ● **PENDING_SURVEY** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **DELETED** ● **DENY** ● **PENDING_PAY** ● **LEASED_LINE_DELIVERY**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
peer_port_type	String	Specifies the peer port type.
peer_provider	String	Specifies the carrier connected to the connection.
order_id	String	Specifies the connection order ID, which is used to support duration-based billing and identify user orders.
product_id	String	Specifies the product ID corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
spec_code	String	Specifies the product specifications corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
period_type	Integer	Specifies whether a connection in a specified order is billed by year or month.
period_num	Integer	Specifies the required service duration of a yearly/monthly connection.
vgw_type	String	Specifies the gateway type required by a direct connection. Default: **default** Enumeration values: ● **default**
lag_id	String	Specifies the ID of the LAG that the connection belongs to.
signed_agreement_status	String	Specifies the signing status of the Direct Connect agreement. Enumeration values: ● **signed**
signed_agreement_time	String	Specifies the time when the line agreement is signed.
enterprise_project_id	String	Specifies the ID of the enterprise project that the connection belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Parameter	Type	Description
locales	**LocalesBody** object	Specifies the region of the connection.
support_feature	Array of strings	Lists the features supported by the connection.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
email	String	Specifies the customer email information.
onestop_product_id	String	Specifies the product ID of a full-service connection. This parameter is used in line sales scenarios.
building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room. This parameter is used in line sales scenarios.
last_onestop_product_id	String	Specifies the product ID of a full-service connection before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
last_building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
modified_bandwidth	Integer	Specifies the new bandwidth after the line bandwidth is changed.
change_mode	Integer	Specifies the status of a renewal change.
onestopdc_status	String	Specifies the status of a full-service connection.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.
auto_renew	Integer	Specifies whether to automatically renew a yearly/monthly subscription.
ratio_95peak	Integer	Specifies the percentage of the minimum bandwidth for 95th percentile billing. Minimum: **0** Maximum: **100**

Table 5-6 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-7 LocalesBody

Parameter	Type	Description
en_us	String	Specifies the region name in English. Minimum: **0** Maximum: **255**
zh_cn	String	Specifies the region name in Chinese. Minimum: **0** Maximum: **255**

Example Requests

Querying details about a connection

```
GET https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/direct-connects/6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a
```

Example Responses

Status code: 200

OK

- The details of the connection are queried.

```
{
  "direct_connect" : {
    "bandwidth" : 100,
    "create_time" : "2018-10-19T09:53:26.000Z",
    "port_type" : "10G",
    "id" : "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "apply_time" : "2018-10-19T09:53:26.000Z",
    "peer_location" : "",
    "peer_port_type" : null,
    "peer_provider" : null,
    "location" : "ExampleLocation",
    "provider" : "ExampleProvider",
    "type" : "standard",
    "status" : "BUILD",
    "description" : "",
    "provider_status" : "ACTIVE",
    "order_id" : "",
    "vlan" : null,
    "device_id" : "172.16.40.2",
    "name" : "direct connect1",
    "admin_state_up" : true,
    "tenant_id" : "6fbe9263116a4b68818cf1edce16bc4f",
    "hosting_id" : null,
    "product_id" : "",
    "vgw_type" : "default",
    "spec_code" : "100ge",
    "charge_mode" : null,
    "support_feature" : [ ],
    "ies_id" : null,
    "reason" : null,
    "email" : "cloud@example.com",
    "onestop_product_id" : null,
    "building_line_product_id" : null,
    "last_building_line_product_id" : null,
    "last_onestop_product_id" : null,
    "modified_bandwidth" : null,
    "change_mode" : null,
    "onestopdc_status" : null,
    "public_border_group" : "center",
    "auto_renew" : 0,
    "ratio_95peak" : null
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ShowDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowDirectConnectRequest request = new ShowDirectConnectRequest();
        request.withFields();
        try {
            ShowDirectConnectResponse response = client.showDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.2 Updating a Connection

Function

This API is used to update a connection, including updating its name and description.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}

Table 5-8 Path Parameters

Parameter	Mandatory	Type	Description
direct_connect_id	Yes	String	Specifies the connection ID. Minimum: **36** Maximum: **36**
project_id	Yes	String	Specifies the project ID.

Request Parameters

Table 5-9 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-10 Request body parameters

Parameter	Mandatory	Type	Description
direct_connect	No	**UpdateDirectConnect** object	Specifies parameters required for updating a connection.

Table 5-11 UpdateDirectConnect

Parameter	Mandatory	Type	Description
name	No	String	Specifies the connection name. Minimum: **0** Maximum: **64**

Parameter	Mandatory	Type	Description
description	No	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	No	Integer	Specifies the bandwidth size of the hosted connection in Mbit/s. Minimum: **2** Maximum: **100000**
peer_location	No	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
status	No	String	Specifies the resource status, which can be **PENDING_PAY** or **APPLY**. Enumeration values: ● **PENDING_PAY** ● **APPLY**
provider_status	No	String	Specifies the carrier status, which can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Response Parameters

Status code: 200

Table 5-12 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
direct_connect	**DirectConnect** object	Specifies parameters for creating a connection.

Table 5-13 DirectConnect

Parameter	Type	Description
id	String	Specifies the connection ID.
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **100000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
device_id	String	Specifies the ID of the device connected to the connection. Minimum: **0** Maximum: **36**

Parameter	Type	Description
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
charge_mode	String	Specifies the billing mode, which can be **prepayment**, **bandwidth**, or **traffic**. Enumeration values: ● **prepayment** ● **bandwidth** ● **traffic**
provider	String	Specifies the line carrier of a connection. It can be China Telecom, China Unicom, China Mobile, and other carriers that provide the lines.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the connection status. The options are as follows: **ACTIVE**: The connection is in the normal state. **DOWN**: The port for the connection is in the down state, which may cause line faults. **BUILD**: Operations related to the connection are being performed. **ERROR**: The connection configuration is incorrect. Contact customer service to rectify the fault. **PENDING_DELETE**: The connection is being deleted. **DELETED**: The connection has been deleted. **APPLY**: A request for a connection is submitted. **DENY**: A site survey is rejected because the customer fails to meet the requirements. **PENDING_PAY**: The order for the connection is to be paid. **PAID**: The order for the connection has been paid. **PENDING_SURVEY**: A site survey is required for the connection. **LEASED_LINE_DELIVERY**: The carrier is delivering the leased line. Enumeration values: ● **BUILD** ● **PAID** ● **APPLY** ● **PENDING_SURVEY** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **DELETED** ● **DENY** ● **PENDING_PAY** ● **LEASED_LINE_DELIVERY**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
peer_port_type	String	Specifies the peer port type.
peer_provider	String	Specifies the carrier connected to the connection.
order_id	String	Specifies the connection order ID, which is used to support duration-based billing and identify user orders.
product_id	String	Specifies the product ID corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
spec_code	String	Specifies the product specifications corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
period_type	Integer	Specifies whether a connection in a specified order is billed by year or month.
period_num	Integer	Specifies the required service duration of a yearly/monthly connection.
vgw_type	String	Specifies the gateway type required by a direct connection. Default: **default** Enumeration values: ● **default**
lag_id	String	Specifies the ID of the LAG that the connection belongs to.
signed_agreement_status	String	Specifies the signing status of the Direct Connect agreement. Enumeration values: ● **signed**
signed_agreement_time	String	Specifies the time when the line agreement is signed.
enterprise_project_id	String	Specifies the ID of the enterprise project that the connection belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Parameter	Type	Description
locales	**LocalesBody** object	Specifies the region of the connection.
support_feature	Array of strings	Lists the features supported by the connection.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
email	String	Specifies the customer email information.
onestop_product_id	String	Specifies the product ID of a full-service connection. This parameter is used in line sales scenarios.
building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room. This parameter is used in line sales scenarios.
last_onestop_product_id	String	Specifies the product ID of a full-service connection before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
last_building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
modified_bandwidth	Integer	Specifies the new bandwidth after the line bandwidth is changed.
change_mode	Integer	Specifies the status of a renewal change.
onestopdc_status	String	Specifies the status of a full-service connection.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.
auto_renew	Integer	Specifies whether to automatically renew a yearly/monthly subscription.
ratio_95peak	Integer	Specifies the percentage of the minimum bandwidth for 95th percentile billing. Minimum: **0** Maximum: **100**

Table 5-14 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-15 LocalesBody

Parameter	Type	Description
en_us	String	Specifies the region name in English. Minimum: **0** Maximum: **255**
zh_cn	String	Specifies the region name in Chinese. Minimum: **0** Maximum: **255**

Example Requests

Updating the name and description of a connection

```
PUT https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/direct-connects/6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a

{
  "direct_connect" : {
    "description" : "",
    "name" : "direct connect1"
  }
}
```

Example Responses

Status code: 200

OK

- The connection is updated.

```
{
  "request_id" : "9a4f4dfc4fb2fc101e65bba07d908535",
  "direct_connect" : {
    "bandwidth" : 100,
    "create_time" : "2018-10-19T09:53:26.000Z",
    "port_type" : "10G",
    "id" : "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "apply_time" : "2018-10-19T09:53:26.000Z",
    "peer_location" : "",
    "peer_port_type" : null,
    "peer_provider" : null,
    "location" : "ExampleLocation",
    "provider" : "ExampleProvider",
    "type" : "standard",
    "status" : "BUILD",
    "description" : "",
    "provider_status" : "ACTIVE",
    "order_id" : "",
    "vlan" : null,
    "device_id" : "172.16.40.2",
    "name" : "direct connect1",
    "admin_state_up" : true,
    "tenant_id" : "6fbe9263116a4b68818cf1edce16bc4f",
    "hosting_id" : null,
    "product_id" : "",
    "vgw_type" : "default",
    "spec_code" : "100ge",
    "charge_mode" : null,
    "support_feature" : [ ],
    "ies_id" : null,
    "reason" : null,
    "email" : "cloud@example.com",
    "onestop_product_id" : null,
    "building_line_product_id" : null,
    "last_building_line_product_id" : null,
    "last_onestop_product_id" : null,
    "modified_bandwidth" : null,
    "change_mode" : null,
    "onestopdc_status" : null,
    "public_border_group" : "center",
    "auto_renew" : 0,
    "ratio_95peak" : null
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the name and description of a connection

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class UpdateDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateDirectConnectRequest request = new UpdateDirectConnectRequest();
        UpdateDirectConnectRequestBody body = new UpdateDirectConnectRequestBody();
        UpdateDirectConnect directConnectbody = new UpdateDirectConnect();
        directConnectbody.withName("direct connect1")
            .withDescription("");
        body.withDirectConnect(directConnectbody);
        request.withBody(body);
        try {
            UpdateDirectConnectResponse response = client.updateDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the name and description of a connection

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateDirectConnectRequest()
        directConnectbody = UpdateDirectConnect(
            name="direct connect1",
            description=""
        )
        request.body = UpdateDirectConnectRequestBody(
            direct_connect=directConnectbody
        )
        response = client.update_direct_connect(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the name and description of a connection

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDirectConnectRequest{}
    nameDirectConnect:= "direct connect1"
    descriptionDirectConnect:= ""
    directConnectbody := &model.UpdateDirectConnect{
        Name: &nameDirectConnect,
        Description: &descriptionDirectConnect,
    }
    request.Body = &model.UpdateDirectConnectRequestBody{
        DirectConnect: directConnectbody,
    }
    response, err := client.UpdateDirectConnect(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.3 Deleting a Connection

Function

This API is used to delete pay-per-use connections only. To delete yearly/monthly connections, you need to first unsubscribe them.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}

Table 5-16 Path Parameters

Parameter	Mandatory	Type	Description
direct_connect_id	Yes	String	Specifies the connection ID. Minimum: **36** Maximum: **36**
project_id	Yes	String	Specifies the project ID.

Request Parameters

Table 5-17 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

None

Example Requests

Deleting a connection

```
DELETE https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/direct-connects/6ecd9cf3-
ca64-46c7-863f-f2eb1b9e838a
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteDirectConnectSolution {

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

```
        // In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteDirectConnectRequest request = new DeleteDirectConnectRequest();
        try {
            DeleteDirectConnectResponse response = client.deleteDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.1.4 Querying the Connection List

Function

This API is used to query all direct connections created by a tenant.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/direct-connects

Table 5-18 Path Parameters

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

Table 5-19 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**

Parameter	Mandatory	Type	Description
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.
hosting_id	No	Array	Specifies operations connection ID by which hosted connections are filtered. Array Length: **0 - 5**
enterprise_project_id	No	Array	Filters resource instances by enterprise project ID. Array Length: **1 - 10**
id	No	Array	Specifies the resource ID by which instances are filtered. Array Length: **1 - 5**
name	No	Array	Specifies the resource name by which instances are filtered. You can specify multiple names. Array Length: **1 - 5**

Request Parameters

Table 5-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-21 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
direct_connects	Array of **DirectConnect** objects	Specifies parameters for creating a connection.
page_info	**PageInfo** object	Specifies the pagination query information.

Table 5-22 DirectConnect

Parameter	Type	Description
id	String	Specifies the connection ID.
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **100000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**

Parameter	Type	Description
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
device_id	String	Specifies the ID of the device connected to the connection. Minimum: **0** Maximum: **36**
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
charge_mode	String	Specifies the billing mode, which can be **prepayment**, **bandwidth**, or **traffic**. Enumeration values: ● **prepayment** ● **bandwidth** ● **traffic**
provider	String	Specifies the line carrier of a connection. It can be China Telecom, China Unicom, China Mobile, and other carriers that provide the lines.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the connection status. The options are as follows: **ACTIVE**: The connection is in the normal state. **DOWN**: The port for the connection is in the down state, which may cause line faults. **BUILD**: Operations related to the connection are being performed. **ERROR**: The connection configuration is incorrect. Contact customer service to rectify the fault. **PENDING_DELETE**: The connection is being deleted. **DELETED**: The connection has been deleted. **APPLY**: A request for a connection is submitted. **DENY**: A site survey is rejected because the customer fails to meet the requirements. **PENDING_PAY**: The order for the connection is to be paid. **PAID**: The order for the connection has been paid. **PENDING_SURVEY**: A site survey is required for the connection. **LEASED_LINE_DELIVERY**: The carrier is delivering the leased line. Enumeration values: ● **BUILD** ● **PAID** ● **APPLY** ● **PENDING_SURVEY** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **DELETED** ● **DENY** ● **PENDING_PAY** ● **LEASED_LINE_DELIVERY**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
peer_port_type	String	Specifies the peer port type.
peer_provider	String	Specifies the carrier connected to the connection.
order_id	String	Specifies the connection order ID, which is used to support duration-based billing and identify user orders.
product_id	String	Specifies the product ID corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
spec_code	String	Specifies the product specifications corresponding to the connection's order, which is used to custom billing policies such as duration-based packages.
period_type	Integer	Specifies whether a connection in a specified order is billed by year or month.
period_num	Integer	Specifies the required service duration of a yearly/monthly connection.
vgw_type	String	Specifies the gateway type required by a direct connection. Default: **default** Enumeration values: ● **default**
lag_id	String	Specifies the ID of the LAG that the connection belongs to.
signed_agreement_status	String	Specifies the signing status of the Direct Connect agreement. Enumeration values: ● **signed**
signed_agreement_time	String	Specifies the time when the line agreement is signed.
enterprise_project_id	String	Specifies the ID of the enterprise project that the connection belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Parameter	Type	Description
locales	**LocalesBody** object	Specifies the region of the connection.
support_feature	Array of strings	Lists the features supported by the connection.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
email	String	Specifies the customer email information.
onestop_product_id	String	Specifies the product ID of a full-service connection. This parameter is used in line sales scenarios.
building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room. This parameter is used in line sales scenarios.
last_onestop_product_id	String	Specifies the product ID of a full-service connection before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
last_building_line_product_id	String	Specifies the product ID of the line resource used in the equipment room before the change. This parameter is used in line sales scenarios and used to save the last record when the line bandwidth is changed.
modified_bandwidth	Integer	Specifies the new bandwidth after the line bandwidth is changed.
change_mode	Integer	Specifies the status of a renewal change.
onestopdc_status	String	Specifies the status of a full-service connection.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.
auto_renew	Integer	Specifies whether to automatically renew a yearly/monthly subscription.
ratio_95peak	Integer	Specifies the percentage of the minimum bandwidth for 95th percentile billing. Minimum: **0** Maximum: **100**

Table 5-23 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-24 LocalesBody

Parameter	Type	Description
en_us	String	Specifies the region name in English. Minimum: **0** Maximum: **255**
zh_cn	String	Specifies the region name in Chinese. Minimum: **0** Maximum: **255**

Table 5-25 PageInfo

Parameter	Type	Description
previous_marker	String	Specifies the marker of the previous page. The value is the resource UUID. Minimum: **0** Maximum: **36**
current_count	Integer	Specifies the number of resources in the current list. Minimum: **0** Maximum: **2000**

Parameter	Type	Description
next_marker	String	Specifies the marker of the next page. The value is the resource UUID. If the value is empty, the resource is on the last page. Minimum: **0** Maximum: **36**

Example Requests

Querying the connection list

```
GET https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/direct-connects
```

Example Responses

Status code: 200

OK

- The connection list is queried.

```
{
  "request_id" : "9a4f4dfc4fb2fc101e65bba07d908535",
  "direct_connects" : [ {
    "bandwidth" : 100,
    "create_time" : "2018-10-19T09:53:26.000Z",
    "port_type" : "10G",
    "id" : "6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a",
    "apply_time" : "2018-10-19T09:53:26.000Z",
    "peer_location" : "",
    "peer_port_type" : null,
    "peer_provider" : null,
    "location" : "ExampleLocation",
    "provider" : "ExampleProvider",
    "type" : "standard",
    "status" : "BUILD",
    "description" : "",
    "provider_status" : "ACTIVE",
    "order_id" : "",
    "vlan" : null,
    "device_id" : "172.16.40.2",
    "name" : "direct connect1",
    "admin_state_up" : true,
    "tenant_id" : "6fbe9263116a4b68818cf1edce16bc4f",
    "hosting_id" : null,
    "product_id" : "",
    "vgw_type" : "default",
    "spec_code" : "100ge",
    "charge_mode" : null,
    "support_feature" : [ ],
    "ies_id" : null,
    "reason" : null,
    "email" : "cloud@example.com",
    "onestop_product_id" : null,
    "building_line_product_id" : null,
    "last_building_line_product_id" : null,
    "last_onestop_product_id" : null,
    "modified_bandwidth" : null,
    "change_mode" : null,
    "onestopdc_status" : null,
    "public_border_group" : "center",
    "auto_renew" : 0,
    "ratio_95peak" : null
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListDirectConnectsSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDirectConnectsRequest request = new ListDirectConnectsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortKey("<sort_key>");
        request.withSortDir();
        request.withHostingId();
        request.withEnterpriseProjectId();
        request.withId();
        request.withName();
        try {
            ListDirectConnectsResponse response = client.listDirectConnects(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListDirectConnectsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields =
        request.sort_key = "<sort_key>"
        request.sort_dir =
        request.hosting_id =
        request.enterprise_project_id =
        request.id =
        request.name =
        response = client.list_direct_connects(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

```
        WithAk(ak).
        WithSk(sk).
        Build()

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDirectConnectsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ListDirectConnects(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.5 Querying the Hosted Connection List

Function

This API is used to query hosted connections created by partners.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/hosted-connects

Table 5-26 Path Parameters

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

Table 5-27 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**
hosting_id	No	Array	Specifies operations connection ID by which hosted connections are filtered. Array Length: **0 - 5**
id	No	Array	Specifies the resource ID by which instances are filtered. Array Length: **1 - 5**

Parameter	Mandatory	Type	Description
name	No	Array	Specifies the resource name by which instances are filtered. You can specify multiple names. Array Length: **1 - 5**

Request Parameters

Table 5-28 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-29 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID of the current operation.
hosted_connects	Array of **HostedDirectConnect** objects	Specifies the parameters for creating a hosted connection.
page_info	**PageInfo** object	Specifies the pagination query information.

Table 5-30 HostedDirectConnect

Parameter	Type	Description
id	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **400000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
provider	String	Specifies the provider of the leased line.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the operating status of the hosted connection. The options are as follows: ● **BUILD**: The hosted connection has been created. ● **ACTIVE**: The associated virtual gateway is normal. ● **DOWN**: The port used by the hosted connection is down, indicating that there may be line faults. ● **ERROR**: The associated virtual gateway is abnormal. ● **PENDING_DELETE**: The hosted connection is being deleted. ● **PENDING_UPDATE**: The hosted connection is being updated. ● **PENDING_CREATE**: The hosted connection is being created. Enumeration values: ● **BUILD** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **PENDING_UPDATE** ● **PENDING_CREATE**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**

Table 5-31 PageInfo

Parameter	Type	Description
previous_marker	String	Specifies the marker of the previous page. The value is the resource UUID. Minimum: **0** Maximum: **36**
current_count	Integer	Specifies the number of resources in the current list. Minimum: **0** Maximum: **2000**
next_marker	String	Specifies the marker of the next page. The value is the resource UUID. If the value is empty, the resource is on the last page. Minimum: **0** Maximum: **36**

Example Requests

Querying the list of hosted connections created by a partner

```
POST https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/hosted-connects
```

Example Responses

Status code: 200

OK

- The list of hosted connections is queried.

```
{
  "request_id" : "a59a3776faa1d055f8124dc7b0977a90",
  "hosted_connects" : [ {
    "id" : "0278b472-ffa5-4eb3-8c0d-979d479f8ef6",
    "name" : "client-dc-faf1",
    "description" : "Hosted Connect",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "hosting_id" : "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
    "vlan" : 441,
    "bandwidth" : 10,
    "location" : "ExampleLocation",
    "peer_location" : "",
    "provider" : "ExampleProvider",
    "type" : "hosted",
    "port_type" : "10G",
    "provider_status" : "ACTIVE",
    "status" : "ACTIVE",
    "apply_time" : "2022-07-13T08:25:38.000Z",
    "admin_state_up" : true,
    "create_time" : "2022-07-13T08:25:38.000Z"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListHostedDirectConnectsSolution {

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListHostedDirectConnectsRequest request = new ListHostedDirectConnectsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortDir();
        request.withSortKey("<sort_key>");
        request.withHostingId();
        request.withId();
        request.withName();
        try {
            ListHostedDirectConnectsResponse response = client.listHostedDirectConnects(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListHostedDirectConnectsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields = 
        request.sort_dir = 
        request.sort_key = "<sort_key>"
        request.hosting_id = 
        request.id = 
        request.name = 
        response = client.list_hosted_direct_connects(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListHostedDirectConnectsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ListHostedDirectConnects(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.6 Creating a Hosted Connection

Function

This API is used by partners to create hosted connections for their users. The creator must have the partner qualification and have an operations connection.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/dcaas/hosted-connects

Table 5-32 Path Parameters

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

Request Parameters

Table 5-33 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-34 Request body parameters

Parameter	Mandatory	Type	Description
hosted_connect	Yes	**CreateHostedDirectConnect** object	Specifies the parameters for creating a hosted connection.

Table 5-35 CreateHostedDirectConnect

Parameter	Mandatory	Type	Description
name	No	String	Specifies the hosted connection name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the hosted connection. Minimum: **0** Maximum: **128**
bandwidth	Yes	Integer	Specifies the bandwidth size of the hosted connection in Mbit/s. Minimum: **2** Maximum: **400000**
hosting_id	Yes	String	Specifies the ID of the operations connection on which the hosted connection is created.
vlan	Yes	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**
resource_tenant_id	Yes	String	Specifies the project ID of the specified tenant for whom a hosted connection is to be created. Minimum: **0** Maximum: **32**

Parameter	Mandatory	Type	Description
peer_location	No	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**

Response Parameters

Status code: 201

Table 5-36 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
hosted_connect	**HostedDirectConnect** object	Specifies the parameters for creating a hosted connection.

Table 5-37 HostedDirectConnect

Parameter	Type	Description
id	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **400000**

Parameter	Type	Description
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
provider	String	Specifies the provider of the leased line.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the operating status of the hosted connection. The options are as follows: ● **BUILD**: The hosted connection has been created. ● **ACTIVE**: The associated virtual gateway is normal. ● **DOWN**: The port used by the hosted connection is down, indicating that there may be line faults. ● **ERROR**: The associated virtual gateway is abnormal. ● **PENDING_DELETE**: The hosted connection is being deleted. ● **PENDING_UPDATE**: The hosted connection is being updated. ● **PENDING_CREATE**: The hosted connection is being created. Enumeration values: ● **BUILD** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **PENDING_UPDATE** ● **PENDING_CREATE**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**

Example Requests

Creating a hosting connection with the bandwidth set to 10 Mbit/s and VLAN to 441

```
POST https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/hosted-connects

{
  "hosted_connect" : {
    "name" : "client-dc-faf1",
    "description" : "Hosted Connect",
    "resource_tenant_id" : "0605768a3300d5762f82c01180692873",
    "hosting_id" : "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
    "vlan" : 441,
    "bandwidth" : 10
  }
}
```

Example Responses

Status code: 201

Created

- The hosted connection is created.

```
{
  "hosted_connect" : {
    "id" : "0278b472-ffa5-4eb3-8c0d-979d479f8ef6",
    "name" : "client-dc-faf1",
    "description" : "Hosted Connect",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "hosting_id" : "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
    "vlan" : 441,
    "bandwidth" : 10,
    "location" : "ExampleLocation",
    "peer_location" : "",
    "provider" : "ExampleProvider",
    "type" : "hosted",
    "port_type" : "10G",
    "provider_status" : "ACTIVE",
    "status" : "ACTIVE",
    "apply_time" : "2022-07-13T08:25:38.000Z",
    "admin_state_up" : true,
    "create_time" : "2022-07-13T08:25:38.000Z"
  },
  "request_id" : "a59a3776faa1d055f8124dc7b0977a90"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a hosting connection with the bandwidth set to 10 Mbit/s and VLAN to 441

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class CreateHostedDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateHostedDirectConnectRequest request = new CreateHostedDirectConnectRequest();
        CreateHostedDirectConnectRequestBody body = new CreateHostedDirectConnectRequestBody();
        CreateHostedDirectConnect hostedConnectbody = new CreateHostedDirectConnect();
        hostedConnectbody.withName("client-dc-faf1")
            .withDescription("Hosted Connect")
            .withBandwidth(10)
            .withHostingId("2cfb53be-b05f-40d5-a2f8-3a59ac383836")
            .withVlan(441)
            .withResourceTenantId("0605768a3300d5762f82c01180692873");
        body.withHostedConnect(hostedConnectbody);
        request.withBody(body);
        try {
            CreateHostedDirectConnectResponse response = client.createHostedDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Creating a hosting connection with the bandwidth set to 10 Mbit/s and VLAN to 441

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateHostedDirectConnectRequest()
        hostedConnectbody = CreateHostedDirectConnect(
            name="client-dc-faf1",
            description="Hosted Connect",
            bandwidth=10,
            hosting_id="2cfb53be-b05f-40d5-a2f8-3a59ac383836",
            vlan=441,
            resource_tenant_id="0605768a3300d5762f82c01180692873"
        )
        request.body = CreateHostedDirectConnectRequestBody(
            hosted_connect=hostedConnectbody
        )
        response = client.create_hosted_direct_connect(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Creating a hosting connection with the bandwidth set to 10 Mbit/s and VLAN to 441

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateHostedDirectConnectRequest{}
    nameHostedConnect:= "client-dc-faf1"
    descriptionHostedConnect:= "Hosted Connect"
    hostedConnectbody := &model.CreateHostedDirectConnect{
        Name: &nameHostedConnect,
        Description: &descriptionHostedConnect,
        Bandwidth: int32(10),
        HostingId: "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
        Vlan: int32(441),
        ResourceTenantId: "0605768a3300d5762f82c01180692873",
    }
    request.Body = &model.CreateHostedDirectConnectRequestBody{
        HostedConnect: hostedConnectbody,
    }
    response, err := client.CreateHostedDirectConnect(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
201	Created

Error Codes

See **Error Codes**.

5.1.7 Querying Details About a Hosted Connection

Function

This API is used to query a hosted connection of a partner.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}

Table 5-38 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
hosted_connect_id	Yes	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**

Table 5-39 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**

Parameter	Mandatory	Type	Description
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**
hosting_id	No	Array	Specifies operations connection ID by which hosted connections are filtered. Array Length: **0 - 5**

Request Parameters

Table 5-40 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-41 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
hosted_connect	**HostedDirectConnect** object	Specifies the parameters for creating a hosted connection.

Table 5-42 HostedDirectConnect

Parameter	Type	Description
id	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **400000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**

Parameter	Type	Description
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
provider	String	Specifies the provider of the leased line.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**
status	String	Specifies the operating status of the hosted connection. The options are as follows: ● **BUILD**: The hosted connection has been created. ● **ACTIVE**: The associated virtual gateway is normal. ● **DOWN**: The port used by the hosted connection is down, indicating that there may be line faults. ● **ERROR**: The associated virtual gateway is abnormal. ● **PENDING_DELETE**: The hosted connection is being deleted. ● **PENDING_UPDATE**: The hosted connection is being updated. ● **PENDING_CREATE**: The hosted connection is being created. Enumeration values: ● **BUILD** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **PENDING_UPDATE** ● **PENDING_CREATE**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.

Parameter	Type	Description
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: ● **1G** ● **10G** ● **40G** ● **100G**
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: ● **standard** ● **hosting** ● **hosted** ● **onestop_standard** ● **onestop_hosted**

Example Requests

Querying details about a hosted connection created by a partner

```
GET https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/hosted-connects/0278b472-
ffa5-4eb3-8c0d-979d479f8ef6
```

Example Responses

Status code: 200

OK

- The details of the hosted connection are queried.

```
{
  "hosted_connect" : {
    "id" : "0278b472-ffa5-4eb3-8c0d-979d479f8ef6",
    "name" : "client-dc-faf1",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "hosting_id" : "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
    "vlan" : 441,
    "bandwidth" : 10,
    "location" : "ExampleLocation",
    "peer_location" : "",
    "provider" : "ExampleProvider",
    "type" : "hosted",
    "port_type" : "10G",
    "provider_status" : "ACTIVE",
    "status" : "ACTIVE",
    "apply_time" : "2022-07-13T08:25:38.000Z",
    "admin_state_up" : true,
    "create_time" : "2022-07-13T08:25:38.000Z"
  },
  "request_id" : "a59a3776faa1d055f8124dc7b0977a90"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ShowHostedDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowHostedDirectConnectRequest request = new ShowHostedDirectConnectRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortDir();
        request.withSortKey("<sort_key>");
        request.withHostingId();
        try {
            ShowHostedDirectConnectResponse response = client.showHostedDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowHostedDirectConnectRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields = 
        request.sort_dir = 
        request.sort_key = "<sort_key>"
        request.hosting_id = 
        response = client.show_hosted_direct_connect(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowHostedDirectConnectRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ShowHostedDirectConnect(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.8 Updating a Hosted Connection

Function

Updating a hosted connection by a partner

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}

Table 5-43 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
hosted_connect_id	Yes	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**

Request Parameters

Table 5-44 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-45 Request body parameters

Parameter	Mandatory	Type	Description
hosted_connect	No	**UpdateHostedDirectConnect** object	Specifies parameters required for updating a hosted connection.

Table 5-46 UpdateHostedDirectConnect

Parameter	Mandatory	Type	Description
name	No	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	No	Integer	Specifies the bandwidth size of the hosted connection in Mbit/s. Minimum: **2** Maximum: **400000**
peer_location	No	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**

Response Parameters

Status code: 200

Table 5-47 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
hosted_connect	**HostedDirectConnect** object	Specifies the parameters for creating a hosted connection.

Table 5-48 HostedDirectConnect

Parameter	Type	Description
id	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the instance belongs to.
name	String	Specifies the connection name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the connection. Minimum: **0** Maximum: **128**
bandwidth	Integer	Specifies the connection bandwidth in Mbit/s. Minimum: **2** Maximum: **400000**
location	String	Specifies information about the Direct Connect location. Minimum: **0** Maximum: **255**
peer_location	String	Specifies the location of the on-premises facility at the other end of the connection, specific to the street or data center name. Minimum: **0** Maximum: **255**
hosting_id	String	Specifies the ID of the operations connection on which the hosted connection is created.
provider	String	Specifies the provider of the leased line.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
vlan	Integer	Specifies the VLAN allocated to the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Type	Description
status	String	Specifies the operating status of the hosted connection. The options are as follows: ● **BUILD**: The hosted connection has been created. ● **ACTIVE**: The associated virtual gateway is normal. ● **DOWN**: The port used by the hosted connection is down, indicating that there may be line faults. ● **ERROR**: The associated virtual gateway is abnormal. ● **PENDING_DELETE**: The hosted connection is being deleted. ● **PENDING_UPDATE**: The hosted connection is being updated. ● **PENDING_CREATE**: The hosted connection is being created. Enumeration values: ● **BUILD** ● **ACTIVE** ● **DOWN** ● **ERROR** ● **PENDING_DELETE** ● **PENDING_UPDATE** ● **PENDING_CREATE**
apply_time	String	Specifies when the connection was requested. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
create_time	String	Specifies when the connection was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used.
provider_status	String	Specifies the status of the carrier's leased line. The status can be **ACTIVE** or **DOWN**. Enumeration values: ● **ACTIVE** ● **DOWN**

Parameter	Type	Description
port_type	String	Specifies the type of the port used by the connection. The value can be **1G**, **10G**, **40G**, or **100G**. Enumeration values: - **1G** - **10G** - **40G** - **100G**
type	String	Specifies the type of a connection. The value can be **standard** (a standard connection), **hosting** (an operations connection) or **hosted** (a hosted connection). Default: **standard** Enumeration values: - **standard** - **hosting** - **hosted** - **onestop_standard** - **onestop_hosted**

Example Requests

Updating the name and description of a hosted connection

```
PUT https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edce16bc4f/dcaas/hosted-connects/0278b472-ffa5-4eb3-8c0d-979d479f8ef6

{
  "hosted_connect" : {
    "name" : "client-dc-faf1",
    "description" : ""
  }
}
```

Example Responses

Status code: 200

OK

- The hosted connection is updated.

```
{
  "hosted_connect" : {
    "id" : "0278b472-ffa5-4eb3-8c0d-979d479f8ef6",
    "name" : "client-dc-faf1",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "hosting_id" : "2cfb53be-b05f-40d5-a2f8-3a59ac383836",
    "vlan" : 441,
    "bandwidth" : 10,
    "location" : "ExampleLocation",
    "peer_location" : "",
    "provider" : "ExampleProvider",
    "type" : "hosted",
    "port_type" : "10G",
    "provider_status" : "ACTIVE",
    "status" : "ACTIVE",
    "apply_time" : "2022-07-13T08:25:38.000Z",
    "admin_state_up" : true,
    "create_time" : "2022-07-13T08:25:38.000Z"
  },
  "request_id" : "a59a3776faa1d055f8124dc7b0977a90"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the name and description of a hosted connection

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class UpdateHostedDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateHostedDirectConnectRequest request = new UpdateHostedDirectConnectRequest();
        UpdateHostedDirectConnectRequestBody body = new UpdateHostedDirectConnectRequestBody();
        UpdateHostedDirectConnect hostedConnectbody = new UpdateHostedDirectConnect();
        hostedConnectbody.withName("client-dc-faf1")
            .withDescription("");
        body.withHostedConnect(hostedConnectbody);
        request.withBody(body);
        try {
            UpdateHostedDirectConnectResponse response = client.updateHostedDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the name and description of a hosted connection

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateHostedDirectConnectRequest()
        hostedConnectbody = UpdateHostedDirectConnect(
            name="client-dc-faf1",
            description=""
        )
        request.body = UpdateHostedDirectConnectRequestBody(
            hosted_connect=hostedConnectbody
        )
        response = client.update_hosted_direct_connect(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the name and description of a hosted connection

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateHostedDirectConnectRequest{}
    nameHostedConnect:= "client-dc-faf1"
    descriptionHostedConnect:= ""
    hostedConnectbody := &model.UpdateHostedDirectConnect{
        Name: &nameHostedConnect,
        Description: &descriptionHostedConnect,
    }
    request.Body = &model.UpdateHostedDirectConnectRequestBody{
        HostedConnect: hostedConnectbody,
    }
    response, err := client.UpdateHostedDirectConnect(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.1.9 Deleting a Hosted Connection

Function

This API is used by partners to delete hosted connections.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}

Table 5-49 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
hosted_connect_id	Yes	String	Specifies the hosted connection ID. Minimum: **36** Maximum: **36**

Request Parameters

Table 5-50 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

None

Example Requests

Deleting a hosted connection

```
DELETE https://{dc_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/dcaas/hosted-connects/
94c2b814-99dc-939a-e811-ae84c61ea3ff
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteHostedDirectConnectSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteHostedDirectConnectRequest request = new DeleteHostedDirectConnectRequest();
        try {
            DeleteHostedDirectConnectResponse response = client.deleteHostedDirectConnect(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.2 Virtual Gateway

5.2.1 Querying Details About a Virtual Gateway

Function

This API is used to query details about a specified virtual gateway.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}

Table 5-51 Path Parameters

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

Table 5-52 Query Parameters

Parameter	Mandatory	Type	Description
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**

Request Parameters

Table 5-53 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-54 Response body parameters

Parameter	Type	Description
virtual_gateway	**VirtualGateway** object	Specifies parameters for creating a virtual gateway.
request_id	String	Specifies the request ID.

Table 5-55 VirtualGateway

Parameter	Type	Description
id	String	Specifies the virtual gateway ID.
vpc_id	String	Specifies the ID of the VPC connected by the virtual gateway.
tenant_id	String	Specifies the ID of the project that the instance belongs to. Minimum: **32** Maximum: **32**
name	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**

Parameter	Type	Description
description	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**
type	String	Specifies the virtual gateway type. The value can only be **default**. Default: **default**
local_ep_group	Array of strings	Specifies the IPv4 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC.
local_ep_group_ipv6	Array of strings	Specifies the IPv6 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC. This is a reserved field.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
status	String	Specifies the virtual gateway status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, or **PENDING_DELETE**.
bgp_asn	Integer	Specifies the local BGP ASN of the virtual gateway. Minimum: **1** Maximum: **4294967295**
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual gateway belongs to. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
redundant_device_id	String	Specifies the ID of the redundant device.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-56 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

Querying a virtual gateway

```
GET https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-gateways/
20082c1b-3c99-48d8-8e8c-116af5d7e9f0
```

Example Responses

Status code: 200

OK

- The details of the virtual gateway associated with a VPC are queried.

```
{
  "virtual_gateway" : {
    "id" : "20082c1b-3c99-48d8-8e8c-116af5d7e9f0",
    "name" : "vgw-c7b22",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fdf03420c",
    "device_id" : "26.151.63.100",
    "redundant_device_id" : "26.152.128.20",
    "type" : "default",
    "status" : "ACTIVE",
    "admin_state_up" : true,
    "bgp_asn" : 64512,
    "local_ep_group" : [ "192.168.1.0/24" ],
    "enterprise_project_id" : "0",
    "public_border_group" : "center"
  },
  "request_id" : "765f7aaf8f2edd0e719de564ef72e2de"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ShowVirtualGatewaySolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowVirtualGatewayRequest request = new ShowVirtualGatewayRequest();
        request.withFields();
        try {
            ShowVirtualGatewayResponse response = client.showVirtualGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.2.2 Updating a Virtual Gateway

Function

This API is used to update a virtual gateway.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}

Table 5-57 Path Parameters

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

Request Parameters

Table 5-58 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-59 Request body parameters

Parameter	Mandatory	Type	Description
virtual_gateway	No	**UpdateVirtualGateway** object	Specifies the virtual gateway to be updated.

Table 5-60 UpdateVirtualGateway

Parameter	Mandatory	Type	Description
name	No	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**
local_ep_group	No	Array of strings	Lists the IPv4 subnets that can be accessed over the virtual gateway. Generally, the list contains the subnet CIDR blocks of the associated VPC. Array Length: **1 - 200**

Parameter	Mandatory	Type	Description
local_ep_group_ipv6	No	Array of strings	Lists the IPv6 subnets that can be accessed using the virtual gateway. Usually, the subnets are those in the VPC associated with the virtual gateway. Array Length: **1 - 50**

Response Parameters

Status code: 200

Table 5-61 Response body parameters

Parameter	Type	Description
virtual_gateway	**VirtualGateway** object	Specifies parameters for creating a virtual gateway.
request_id	String	Specifies the request ID.

Table 5-62 VirtualGateway

Parameter	Type	Description
id	String	Specifies the virtual gateway ID.
vpc_id	String	Specifies the ID of the VPC connected by the virtual gateway.
tenant_id	String	Specifies the ID of the project that the instance belongs to. Minimum: **32** Maximum: **32**
name	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**

Parameter	Type	Description
type	String	Specifies the virtual gateway type. The value can only be **default**. Default: **default**
local_ep_group	Array of strings	Specifies the IPv4 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC.
local_ep_group_ipv6	Array of strings	Specifies the IPv6 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC. This is a reserved field.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
status	String	Specifies the virtual gateway status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, or **PENDING_DELETE**.
bgp_asn	Integer	Specifies the local BGP ASN of the virtual gateway. Minimum: **1** Maximum: **4294967295**
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual gateway belongs to. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
redundant_device_id	String	Specifies the ID of the redundant device.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-63 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

Updating the name and description of a virtual gateway and changing its IPv4 subnet to 192.168.3.0/24

```
PUT https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-gateways/
20082c1b-3c99-48d8-8e8c-116af5d7e9f0

{
  "virtual_gateway" : {
    "name" : "update-vgw-c7b22",
    "description" : "",
    "local_ep_group" : [ "192.168.3.0/24" ]
  }
}
```

Example Responses

Status code: 200

OK

- The virtual gateway associated with a VPC is updated.

```
{
  "virtual_gateway" : {
    "id" : "20082c1b-3c99-48d8-8e8c-116af5d7e9f0",
    "name" : "update-vgw-c7b22",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fdf03420c",
    "device_id" : "26.151.63.100",
    "redundant_device_id" : "26.152.128.20",
    "type" : "default",
    "status" : "ACTIVE",
    "admin_state_up" : true,
    "bgp_asn" : 64512,
    "local_ep_group" : [ "192.168.3.0/24" ],
    "enterprise_project_id" : "0",
    "public_border_group" : "center"
  },
  "request_id" : "765f7aaf8f2edd0e719de564ef72e2de"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the name and description of a virtual gateway and changing its IPv4 subnet to 192.168.3.0/24

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class UpdateVirtualGatewaySolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateVirtualGatewayRequest request = new UpdateVirtualGatewayRequest();
        UpdateVirtualGatewayRequestBody body = new UpdateVirtualGatewayRequestBody();
        List<String> listVirtualGatewayLocalEpGroup = new ArrayList<>();
        listVirtualGatewayLocalEpGroup.add("192.168.3.0/24");
        UpdateVirtualGateway virtualGatewaybody = new UpdateVirtualGateway();
        virtualGatewaybody.withName("update-vgw-c7b22")
            .withDescription("")
            .withLocalEpGroup(listVirtualGatewayLocalEpGroup);
        body.withVirtualGateway(virtualGatewaybody);
        request.withBody(body);
        try {
            UpdateVirtualGatewayResponse response = client.updateVirtualGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the name and description of a virtual gateway and changing its IPv4 subnet to 192.168.3.0/24

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateVirtualGatewayRequest()
        listLocalEpGroupVirtualGateway = [
            "192.168.3.0/24"
        ]
        virtualGatewaybody = UpdateVirtualGateway(
            name="update-vgw-c7b22",
            description="",
            local_ep_group=listLocalEpGroupVirtualGateway
        )
        request.body = UpdateVirtualGatewayRequestBody(
            virtual_gateway=virtualGatewaybody
        )
        response = client.update_virtual_gateway(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the name and description of a virtual gateway and changing its IPv4 subnet to 192.168.3.0/24

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVirtualGatewayRequest{}
    var listLocalEpGroupVirtualGateway = []string{
        "192.168.3.0/24",
    }
    nameVirtualGateway:= "update-vgw-c7b22"
    descriptionVirtualGateway:= ""
    virtualGatewaybody := &model.UpdateVirtualGateway{
        Name: &nameVirtualGateway,
        Description: &descriptionVirtualGateway,
        LocalEpGroup: &listLocalEpGroupVirtualGateway,
    }
    request.Body = &model.UpdateVirtualGatewayRequestBody{
        VirtualGateway: virtualGatewaybody,
    }
    response, err := client.UpdateVirtualGateway(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.2.3 Deleting a Virtual Gateway

Function

The API is used to delete a specified virtual gateway.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}

Table 5-64 Path Parameters

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

Request Parameters

Table 5-65 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

None

Example Requests

Deleting a virtual gateway

```
DELETE https://{dc_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/dcaas/virtual-gateways/
20082c1b-3c99-48d8-8e8c-116af5d7e9f0
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteVirtualGatewaySolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteVirtualGatewayRequest request = new DeleteVirtualGatewayRequest();
        try {
            DeleteVirtualGatewayResponse response = client.deleteVirtualGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.2.4 Querying Virtual Gateways

Function

This API is used to query virtual gateways.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/virtual-gateways

Table 5-66 Path Parameters

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

Table 5-67 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**

Parameter	Mandatory	Type	Description
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**
id	No	Array	Specifies the resource ID by which instances are filtered. Array Length: **1 - 5**
enterprise_project_id	No	Array	Filters resource instances by enterprise project ID. Array Length: **1 - 10**
vpc_id	No	Array	Specifies the VPC ID by which virtual gateways are filtered.

Request Parameters

Table 5-68 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-69 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
virtual_gateways	Array of **VirtualGateway** objects	Specifies parameters for creating a virtual gateway.
page_info	**PageInfo** object	Specifies the pagination query information.

Table 5-70 VirtualGateway

Parameter	Type	Description
id	String	Specifies the virtual gateway ID.
vpc_id	String	Specifies the ID of the VPC connected by the virtual gateway.
tenant_id	String	Specifies the ID of the project that the instance belongs to. Minimum: **32** Maximum: **32**

Parameter	Type	Description
name	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**
type	String	Specifies the virtual gateway type. The value can only be **default**. Default: **default**
local_ep_group	Array of strings	Specifies the IPv4 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC.
local_ep_group_ipv6	Array of strings	Specifies the IPv6 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC. This is a reserved field.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
status	String	Specifies the virtual gateway status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, or **PENDING_DELETE**.
bgp_asn	Integer	Specifies the local BGP ASN of the virtual gateway. Minimum: **1** Maximum: **4294967295**
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual gateway belongs to. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
redundant_device_id	String	Specifies the ID of the redundant device.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.

Parameter	Type	Description
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-71 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-72 PageInfo

Parameter	Type	Description
previous_marker	String	Specifies the marker of the previous page. The value is the resource UUID. Minimum: **0** Maximum: **36**
current_count	Integer	Specifies the number of resources in the current list. Minimum: **0** Maximum: **2000**
next_marker	String	Specifies the marker of the next page. The value is the resource UUID. If the value is empty, the resource is on the last page. Minimum: **0** Maximum: **36**

Example Requests

Querying the virtual gateway list

```
GET https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-gateways
```

Example Responses

Status code: 200

OK

- The list of virtual gateways associated with VPCs is queried.

```
{
  "virtual_gateways" : [ {
    "id" : "20082c1b-3c99-48d8-8e8c-116af5d7e9f0",
    "name" : "vgw-c7b22",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fdf03420c",
    "device_id" : "26.151.63.100",
    "redundant_device_id" : "26.152.128.20",
    "type" : "default",
    "status" : "ACTIVE",
    "admin_state_up" : true,
    "bgp_asn" : 64512,
    "local_ep_group" : [ "192.168.1.0/24" ],
    "enterprise_project_id" : "0",
    "public_border_group" : "center"
  } ],
  "request_id" : "765f7aaf8f2edd0e719de564ef72e2de"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListVirtualGatewaysSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListVirtualGatewaysRequest request = new ListVirtualGatewaysRequest();
        request.withVpcId();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortDir();
        request.withSortKey("<sort_key>");
        request.withId();
        request.withEnterpriseProjectId();
        try {
            ListVirtualGatewaysResponse response = client.listVirtualGateways(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListVirtualGatewaysRequest()
        request.vpc_id = 
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields = 
        request.sort_dir = 
        request.sort_key = "<sort_key>"
        request.id = 
        request.enterprise_project_id = 
        response = client.list_virtual_gateways(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListVirtualGatewaysRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ListVirtualGateways(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.2.5 Creating a Virtual Gateway

Function

This API is used to create a virtual gateway.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/dcaas/virtual-gateways

Table 5-73 Path Parameters

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

Request Parameters

Table 5-74 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-75 Request body parameters

Parameter	Mandatory	Type	Description
virtual_gateway	No	**CreateVirtualGateway** object	Specifies parameters for creating a virtual gateway.

Table 5-76 CreateVirtualGateway

Parameter	Mandatory	Type	Description
vpc_id	Yes	String	Specifies the ID of the VPC that the virtual gateway is associated with.
name	No	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**
local_ep_group	Yes	Array of strings	Lists the IPv4 subnets that can be accessed over the virtual gateway. Generally, the list contains the subnet CIDR blocks of the associated VPC.
local_ep_group_ipv6	No	Array of strings	Specifies the IPv6 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC. This is a reserved field.
bgp_asn	No	Integer	Specifies the local BGP autonomous system number (ASN) of the virtual gateway. Minimum: **1** Maximum: **4294967295**
enterprise_project_id	No	String	Specifies the ID of the enterprise project that the virtual gateway belongs to. Minimum: **36** Maximum: **36**
tags	No	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-77 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	No	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Response Parameters

Status code: 201

Table 5-78 Response body parameters

Parameter	Type	Description
virtual_gateway	**VirtualGateway** object	Specifies parameters for creating a virtual gateway.
request_id	String	Specifies the request ID.

Table 5-79 VirtualGateway

Parameter	Type	Description
id	String	Specifies the virtual gateway ID.
vpc_id	String	Specifies the ID of the VPC connected by the virtual gateway.
tenant_id	String	Specifies the ID of the project that the instance belongs to. Minimum: **32** Maximum: **32**

Parameter	Type	Description
name	String	Specifies the virtual gateway name. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual gateway. Minimum: **0** Maximum: **128**
type	String	Specifies the virtual gateway type. The value can only be **default**. Default: **default**
local_ep_group	Array of strings	Specifies the IPv4 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC.
local_ep_group_ipv6	Array of strings	Specifies the IPv6 subnets connected by the virtual gateway, which is usually the CIDR blocks of a VPC. This is a reserved field.
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**. Default: **true**
status	String	Specifies the virtual gateway status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, or **PENDING_DELETE**.
bgp_asn	Integer	Specifies the local BGP ASN of the virtual gateway. Minimum: **1** Maximum: **4294967295**
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual gateway belongs to. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
redundant_device_id	String	Specifies the ID of the redundant device.
public_border_group	String	Specifies the public border group of the AZ, indicating whether the site is a HomeZones site.

Parameter	Type	Description
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-80 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

- Creating a virtual gateway that will be used to access a VPC, with the local BGP ASN set to 64512 and the IPv4 subnet to 192.168.1.0/24

```
POST https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-gateways

{
  "virtual_gateway" : {
    "name" : "vgw-c7b22",
    "description" : "",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fdf03420c",
    "bgp_asn" : 64512,
    "local_ep_group" : [ "192.168.1.0/24" ]
  }
}
```

Example Responses

Status code: 201

Created

- The virtual gateway for accessing the VPC is created

```
{
  "virtual_gateway" : {
    "id" : "20082c1b-3c99-48d8-8e8c-116af5d7e9f0",
    "name" : "vgw-c7b22",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fdf03420c",
    "device_id" : "26.151.63.100",
    "redundant_device_id" : "26.152.128.20",
    "type" : "default",
    "status" : "ACTIVE",
    "admin_state_up" : true,
    "bgp_asn" : 64512,
    "local_ep_group" : [ "192.168.1.0/24" ],
    "enterprise_project_id" : "0",
    "public_border_group" : "center"
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Creating a virtual gateway that will be used to access a VPC, with the local BGP ASN set to 64512 and the IPv4 subnet to 192.168.1.0/24

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class CreateVirtualGatewaySolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateVirtualGatewayRequest request = new CreateVirtualGatewayRequest();
        CreateVirtualGatewayRequestBody body = new CreateVirtualGatewayRequestBody();
        List<String> listVirtualGatewayLocalEpGroup = new ArrayList<>();
        listVirtualGatewayLocalEpGroup.add("192.168.1.0/24");
        CreateVirtualGateway virtualGatewaybody = new CreateVirtualGateway();
        virtualGatewaybody.withVpcId("6592c28e-95d7-4b0a-9f61-004fdf03420c")
            .withName("vgw-c7b22")
            .withDescription("")
            .withLocalEpGroup(listVirtualGatewayLocalEpGroup)
            .withBgpAsn(64512);
        body.withVirtualGateway(virtualGatewaybody);
        request.withBody(body);
        try {
            CreateVirtualGatewayResponse response = client.createVirtualGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Creating a virtual gateway that will be attached to an enterprise router and set the BGP ASN to 64512

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class CreateVirtualGatewaySolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateVirtualGatewayRequest request = new CreateVirtualGatewayRequest();
        CreateVirtualGatewayRequestBody body = new CreateVirtualGatewayRequestBody();
        CreateVirtualGateway virtualGatewaybody = new CreateVirtualGateway();
        virtualGatewaybody.withName("vgw-er")
            .withDescription("")
            .withBgpAsn(64512);
        body.withVirtualGateway(virtualGatewaybody);
        request.withBody(body);
        try {
            CreateVirtualGatewayResponse response = client.createVirtualGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Creating a virtual gateway that will be used to access a VPC, with the local BGP ASN set to 64512 and the IPv4 subnet to 192.168.1.0/24

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateVirtualGatewayRequest()
        listLocalEpGroupVirtualGateway = [
            "192.168.1.0/24"
        ]
        virtualGatewaybody = CreateVirtualGateway(
            vpc_id="6592c28e-95d7-4b0a-9f61-004fdf03420c",
            name="vgw-c7b22",
            description="",
            local_ep_group=listLocalEpGroupVirtualGateway,
            bgp_asn=64512
        )
        request.body = CreateVirtualGatewayRequestBody(
            virtual_gateway=virtualGatewaybody
        )
        response = client.create_virtual_gateway(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Creating a virtual gateway that will be attached to an enterprise router and set the BGP ASN to 64512

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

```
environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateVirtualGatewayRequest()
        virtualGatewaybody = CreateVirtualGateway(
            name="vgw-er",
            description="",
            bgp_asn=64512
        )
        request.body = CreateVirtualGatewayRequestBody(
            virtual_gateway=virtualGatewaybody
        )
        response = client.create_virtual_gateway(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Creating a virtual gateway that will be used to access a VPC, with the local BGP ASN set to 64512 and the IPv4 subnet to 192.168.1.0/24

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateVirtualGatewayRequest{}
    var listLocalEpGroupVirtualGateway = []string{
        "192.168.1.0/24",
    }
    nameVirtualGateway:= "vgw-c7b22"
    descriptionVirtualGateway:= ""
    bgpAsnVirtualGateway:= int32(64512)
    virtualGatewaybody := &model.CreateVirtualGateway{
        VpcId: "6592c28e-95d7-4b0a-9f61-004fdf03420c",
        Name: &nameVirtualGateway,
        Description: &descriptionVirtualGateway,
        LocalEpGroup: listLocalEpGroupVirtualGateway,
        BgpAsn: &bgpAsnVirtualGateway,
    }
    request.Body = &model.CreateVirtualGatewayRequestBody{
        VirtualGateway: virtualGatewaybody,
    }
    response, err := client.CreateVirtualGateway(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Creating a virtual gateway that will be attached to an enterprise router and set the BGP ASN to 64512

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateVirtualGatewayRequest{}
    nameVirtualGateway:= "vgw-er"
    descriptionVirtualGateway:= ""
    bgpAsnVirtualGateway:= int32(64512)
    virtualGatewaybody := &model.CreateVirtualGateway{
        Name: &nameVirtualGateway,
        Description: &descriptionVirtualGateway,
        BgpAsn: &bgpAsnVirtualGateway,
    }
    request.Body = &model.CreateVirtualGatewayRequestBody{
        VirtualGateway: virtualGatewaybody,
    }
    response, err := client.CreateVirtualGateway(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
```

```
  }
}
```

More

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

Status Codes

Status Code	Description
201	Created

Error Codes

See **Error Codes**.

5.3 Virtual Interface

5.3.1 Querying Details About a Virtual Interface

Function

This API is used to querying details about a virtual interface.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}

Table 5-81 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
virtual_interface_id	Yes	String	Specifies the virtual interface ID. Minimum: **36** Maximum: **36**

Table 5-82 Query Parameters

Parameter	Mandatory	Type	Description
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**

Request Parameters

Table 5-83 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-84 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
virtual_interface	**VirtualInterface** object	Specifies parameters for creating a virtual interface.

Table 5-85 VirtualInterface

Parameter	Type	Description
id	String	Specifies the virtual interface ID. Maximum: **36**
name	String	Specifies the virtual interface name. Maximum: **64**

Parameter	Type	Description
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**.
bandwidth	Integer	Specifies the virtual interface bandwidth. Minimum: **2** Maximum: **2147483647**
create_time	String	Specifies the time when the virtual interface was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used. Maximum: **255**
description	String	Provides supplementary information about the virtual interface. Maximum: **128**
direct_connect_id	String	Specifies the connection ID. Maximum: **36**
service_type	String	Specifies the gateway type, which can be **VGW**, **GDGW**, or **LGW**. Enumeration values: ● **VGW** ● **GDGW** ● **LGW**
status	String	Specifies the operating status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, **PENDING_DELETE**, **DELETED**, **AUTHORIZATION**, or **REJECTED**.
tenant_id	String	Specifies the project ID. Minimum: **32** Maximum: **32**
type	String	Specifies the type of the virtual interface. The value is **private**. Default: **private** Maximum: **255** Enumeration values: ● **private** ● **public**
vgw_id	String	Specifies the virtual gateway ID. Minimum: **36** Maximum: **36**

Parameter	Type	Description
vlan	Integer	Specifies the VLAN for connecting to the user gateway. The value ranges from **0** to **3999**. Minimum: **0** Maximum: **3999**
route_limit	Integer	Specifies the remote subnet route configurations of the virtual interface. Minimum: **1** Maximum: **200** Default: **50**
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable Bidirectional Forwarding Detection (BFD). The value can be **true** or **false**.
lag_id	String	Specifies the ID of the LAG associated with the virtual interface. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual interface belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**
local_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
rate_limit	Boolean	Specifies whether rate limiting is enabled on a virtual interface.

Parameter	Type	Description
address_family	String	Specifies the address family of the virtual interface, which can be **IPv4** or **IPv6**. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
local_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
lgw_id	String	Specifies the ID of the local gateway, which is used in IES scenarios.
gateway_id	String	Specifies the ID of the gateway associated with the virtual interface.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
bgp_route_limit	Integer	Specifies the BGP route configuration.
priority	String	Specifies the priority of a virtual interface. The value can be **normal** or **low**. If the priorities are the same, the virtual interfaces work in load balancing mode. If the priorities are different, the virtual interfaces work in active/standby pairs. Outbound traffic is preferentially forwarded to the normal virtual interface with a higher priority. This option is only supported by virtual interfaces that use BGP routing. Default: **normal** Enumeration values: ● **normal** ● **low**

Parameter	Type	Description
vif_peers	Array of **VifPeer** objects	Provides information about virtual interface peers.
extend_attribute	**VifExtendAttribute** object	Provides extended parameter information.

Table 5-86 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-87 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**

Parameter	Type	Description
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**

Parameter	Type	Description
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Table 5-88 VifExtendAttribute

Parameter	Type	Description
ha_type	String	Specifies the availability detection type of the virtual interface. Enumeration values: ● **nqa** ● **bfd**
ha_mode	String	Specifies the availability detection mode. Enumeration values: ● **auto_single** ● **auto_multi** ● **static_single** ● **static_multi** ● **enhance_nqa**
detect_multiplier	Integer	Specifies the number of detection retries. Default: **5**
min_rx_interval	Integer	Specifies the interval for receiving detection packets. Default: **1000**
min_tx_interval	Integer	Specifies the interval for sending detection packets. Default: **1000**
remote_disclaim	Integer	Specifies the remote identifier of the static BFD session.
local_disclaim	Integer	Specifies the local identifier of the static BFD session.

Example Requests

Querying a virtual interface

```
POST https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-interfaces/0d0fdf63-f2c4-491c-8866-d504796189be
```

Example Responses

Status code: 200

OK

```
{
  "virtual_interface" : {
    "id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
    "name" : "vif-0819",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "direct_connect_id" : "4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
    "vgw_id" : "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
    "type" : "private",
    "service_type" : "VGW",
    "vlan" : 332,
    "bandwidth" : 2,
    "status" : "ACTIVE",
    "create_time" : "2022-08-19T11:28:06.000Z",
    "admin_state_up" : true,
    "enable_bfd" : false,
    "route_limit" : 50,
    "enable_nqa" : false,
    "local_gateway_v4_ip" : "1.1.1.1/30",
    "remote_gateway_v4_ip" : "1.1.1.2/30",
    "ies_id" : null,
    "reason" : null,
    "rate_limit" : false,
    "address_family" : "ipv4",
    "local_gateway_v6_ip" : null,
    "remote_gateway_v6_ip" : null,
    "lgw_id" : null,
    "lag_id" : null,
    "gateway_id" : null,
    "remote_ep_group" : [ "1.1.2.0/30" ],
    "service_ep_group" : [ ],
    "bgp_route_limit" : 100,
    "priority" : "normal",
    "vif_peers" : [ {
      "id" : "c768eb52-12a8-4859-9b43-81194643040c",
      "tenant_id" : "0605768a3300d5762f82c01180692873",
      "name" : "vif-0819",
      "description" : "",
      "address_family" : "ipv4",
      "local_gateway_ip" : "1.1.1.1/30",
      "remote_gateway_ip" : "1.1.1.2/30",
      "route_mode" : "static",
      "bgp_asn" : null,
      "bgp_md5" : null,
      "device_id" : "18.9.215.131",
      "bgp_route_limit" : 100,
      "bgp_status" : null,
      "status" : "ACTIVE",
      "vif_id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
      "receive_route_num" : -1,
      "remote_ep_group" : [ "1.1.2.0/30" ],
      "service_ep_group" : null,
      "enable_bfd" : false,
      "enable_nqa" : false
    } ],
    "enterprise_project_id" : "0"
  },
  "request_id" : "5633df7af874576d819a481c76673236"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ShowVirtualInterfaceSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowVirtualInterfaceRequest request = new ShowVirtualInterfaceRequest();
        request.withFields();
        try {
            ShowVirtualInterfaceResponse response = client.showVirtualInterface(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.3.2 Updating a Virtual Interface

Function

This API is used to update a virtual interface.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}

Table 5-89 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
virtual_interface_id	Yes	String	Specifies the virtual interface ID. Minimum: **36** Maximum: **36**

Request Parameters

Table 5-90 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-91 Request body parameters

Parameter	Mandatory	Type	Description
virtual_interface	Yes	**UpdateVirtualInterface** object	Virtual interface objects to be updated

Table 5-92 UpdateVirtualInterface

Parameter	Mandatory	Type	Description
name	No	String	Specifies the virtual interface name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the virtual interface. Minimum: **0** Maximum: **128**
bandwidth	No	Integer	Specifies the virtual interface bandwidth configuration. Minimum: **2** Maximum: **2147483647**

Parameter	Mandatory	Type	Description
remote_ep_group	No	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	No	Array of strings	Specifies the local endpoint group that is connected to the public network over a connection.
enable_bfd	No	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.
enable_nqa	No	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
status	No	String	Confirms the virtual interfaces created by other users. The value can be **ACCEPTED** or **REJECTED**. Enumeration values: ● **ACCEPTED** ● **REJECTED**

Response Parameters

Status code: 200

Table 5-93 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
virtual_interface	**VirtualInterface** object	Specifies parameters for creating a virtual interface.

Table 5-94 VirtualInterface

Parameter	Type	Description
id	String	Specifies the virtual interface ID. Maximum: **36**

Parameter	Type	Description
name	String	Specifies the virtual interface name. Maximum: **64**
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**.
bandwidth	Integer	Specifies the virtual interface bandwidth. Minimum: **2** Maximum: **2147483647**
create_time	String	Specifies the time when the virtual interface was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used. Maximum: **255**
description	String	Provides supplementary information about the virtual interface. Maximum: **128**
direct_connect_id	String	Specifies the connection ID. Maximum: **36**
service_type	String	Specifies the gateway type, which can be **VGW**, **GDGW**, or **LGW**. Enumeration values: ● **VGW** ● **GDGW** ● **LGW**
status	String	Specifies the operating status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, **PENDING_DELETE**, **DELETED**, **AUTHORIZATION**, or **REJECTED**.
tenant_id	String	Specifies the project ID. Minimum: **32** Maximum: **32**
type	String	Specifies the type of the virtual interface. The value is **private**. Default: **private** Maximum: **255** Enumeration values: ● **private** ● **public**

Parameter	Type	Description
vgw_id	String	Specifies the virtual gateway ID. Minimum: **36** Maximum: **36**
vlan	Integer	Specifies the VLAN for connecting to the user gateway. The value ranges from **0** to **3999**. Minimum: **0** Maximum: **3999**
route_limit	Integer	Specifies the remote subnet route configurations of the virtual interface. Minimum: **1** Maximum: **200** Default: **50**
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable Bidirectional Forwarding Detection (BFD). The value can be **true** or **false**.
lag_id	String	Specifies the ID of the LAG associated with the virtual interface. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual interface belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**
local_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
ies_id	String	Specifies the ID of an IES edge site.

Parameter	Type	Description
reason	String	Displays error information if the status of a line is **Error**.
rate_limit	Boolean	Specifies whether rate limiting is enabled on a virtual interface.
address_family	String	Specifies the address family of the virtual interface, which can be **IPv4** or **IPv6**. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
local_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
lgw_id	String	Specifies the ID of the local gateway, which is used in IES scenarios.
gateway_id	String	Specifies the ID of the gateway associated with the virtual interface.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
bgp_route_limit	Integer	Specifies the BGP route configuration.

Parameter	Type	Description
priority	String	Specifies the priority of a virtual interface. The value can be **normal** or **low**. If the priorities are the same, the virtual interfaces work in load balancing mode. If the priorities are different, the virtual interfaces work in active/ standby pairs. Outbound traffic is preferentially forwarded to the normal virtual interface with a higher priority. This option is only supported by virtual interfaces that use BGP routing. Default: **normal** Enumeration values: ● **normal** ● **low**
vif_peers	Array of **VifPeer** objects	Provides information about virtual interface peers.
extend_attribute	**VifExtendAttribute** object	Provides extended parameter information.

Table 5-95 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-96 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**

Parameter	Type	Description
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.

Parameter	Type	Description
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Table 5-97 VifExtendAttribute

Parameter	Type	Description
ha_type	String	Specifies the availability detection type of the virtual interface. Enumeration values: ● **nqa** ● **bfd**
ha_mode	String	Specifies the availability detection mode. Enumeration values: ● **auto_single** ● **auto_multi** ● **static_single** ● **static_multi** ● **enhance_nqa**
detect_multiplier	Integer	Specifies the number of detection retries. Default: **5**

Parameter	Type	Description
min_rx_interval	Integer	Specifies the interval for receiving detection packets. Default: **1000**
min_tx_interval	Integer	Specifies the interval for sending detection packets. Default: **1000**
remote_disclaim	Integer	Specifies the remote identifier of the static BFD session.
local_disclaim	Integer	Specifies the local identifier of the static BFD session.

Example Requests

Updating the name and description of a virtual interface and changing its bandwidth to 2 Mbit/s

```
PUT https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-interfaces/0d0fdf63-f2c4-491c-8866-d504796189be

{
  "virtual_interface" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "bandwidth" : 2
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "virtual_interface" : {
    "id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
    "name" : "vif-0819",
    "description" : "mytest",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "direct_connect_id" : "4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
    "vgw_id" : "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
    "type" : "private",
    "service_type" : "VGW",
    "vlan" : 332,
    "bandwidth" : 2,
    "status" : "ACTIVE",
    "create_time" : "2022-08-19T11:28:06.000Z",
    "admin_state_up" : true,
    "enable_bfd" : false,
    "route_limit" : 50,
    "enable_nqa" : false,
    "local_gateway_v4_ip" : "1.1.1.1/30",
    "remote_gateway_v4_ip" : "1.1.1.2/30",
    "ies_id" : null,
    "reason" : null,
    "rate_limit" : false,
    "address_family" : "ipv4",
    "local_gateway_v6_ip" : null,
    "remote_gateway_v6_ip" : null,
    "lgw_id" : null,
    "lag_id" : null,
    "gateway_id" : null,
    "remote_ep_group" : [ "1.1.2.0/30" ],
    "service_ep_group" : [ ],
    "bgp_route_limit" : 100,
    "priority" : "normal",
    "vif_peers" : [ {
      "id" : "c768eb52-12a8-4859-9b43-81194643040c",
      "tenant_id" : "0605768a3300d5762f82c01180692873",
      "name" : "vif-0819",
      "description" : "",
      "address_family" : "ipv4",
      "local_gateway_ip" : "1.1.1.1/30",
      "remote_gateway_ip" : "1.1.1.2/30",
      "route_mode" : "static",
      "bgp_asn" : null,
      "bgp_md5" : null,
      "device_id" : "18.9.215.131",
      "bgp_route_limit" : 100,
      "bgp_status" : null,
      "status" : "ACTIVE",
      "vif_id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
      "receive_route_num" : -1,
      "remote_ep_group" : [ "1.1.2.0/30" ],
      "service_ep_group" : null,
      "enable_bfd" : false,
      "enable_nqa" : false
    } ],
    "enterprise_project_id" : "0"
  },
  "request_id" : "5633df7af874576d819a481c76673236"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the name and description of a virtual interface and changing its bandwidth to 2 Mbit/s

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class UpdateVirtualInterfaceSolution {

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

```
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateVirtualInterfaceRequest request = new UpdateVirtualInterfaceRequest();
        UpdateVirtualInterfaceRequestBody body = new UpdateVirtualInterfaceRequestBody();
        UpdateVirtualInterface virtualInterfacebody = new UpdateVirtualInterface();
        virtualInterfacebody.withName("vif-0819")
            .withDescription("mytest")
            .withBandwidth(2);
        body.withVirtualInterface(virtualInterfacebody);
        request.withBody(body);
        try {
            UpdateVirtualInterfaceResponse response = client.updateVirtualInterface(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the name and description of a virtual interface and changing its bandwidth to 2 Mbit/s

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateVirtualInterfaceRequest()
        virtualInterfacebody = UpdateVirtualInterface(
            name="vif-0819",
            description="mytest",
            bandwidth=2
        )
        request.body = UpdateVirtualInterfaceRequestBody(
            virtual_interface=virtualInterfacebody
        )
        response = client.update_virtual_interface(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the name and description of a virtual interface and changing its bandwidth to 2 Mbit/s

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVirtualInterfaceRequest{}
    nameVirtualInterface:= "vif-0819"
    descriptionVirtualInterface:= "mytest"
    bandwidthVirtualInterface:= int32(2)
    virtualInterfacebody := &model.UpdateVirtualInterface{
        Name: &nameVirtualInterface,
        Description: &descriptionVirtualInterface,
        Bandwidth: &bandwidthVirtualInterface,
    }
    request.Body = &model.UpdateVirtualInterfaceRequestBody{
        VirtualInterface: virtualInterfacebody,
    }
    response, err := client.UpdateVirtualInterface(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.3.3 Deleting a Virtual Interface

Function

This API is used to delete a virtual interface.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}

Table 5-98 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
virtual_interface_id	Yes	String	Specifies the virtual interface ID. Minimum: **36** Maximum: **36**

Request Parameters

Table 5-99 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

None

Example Requests

Deleting a virtual interface

```
DELETE https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-interfaces/0d0fdf63-f2c4-491c-8866-d504796189be
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteVirtualInterfaceSolution {

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

```
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteVirtualInterfaceRequest request = new DeleteVirtualInterfaceRequest();
        try {
            DeleteVirtualInterfaceResponse response = client.deleteVirtualInterface(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.3.4 Querying the Virtual Interface List

Function

This API is used to query all virtual interfaces of a tenant.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/virtual-interfaces

Table 5-100 Path Parameters

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

Table 5-101 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.

Parameter	Mandatory	Type	Description
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**
enterprise_project_id	No	Array	Filters resource instances by enterprise project ID. Array Length: **1 - 10**
id	No	Array	Specifies the resource ID by which instances are filtered. Array Length: **1 - 5**
status	No	Array	Specifies the status by which instances are filtered. Array Length: **1 - 5**
direct_connect_id	No	Array	Specifies the connection ID by which connections are filtered and queried. Array Length: **1 - 5**
vgw_id	No	Array	Specifies the virtual gateway ID by which virtual gateways are filtered and queried. Array Length: **1 - 5**

Request Parameters

Table 5-102 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-103 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
virtual_interfaces	Array of **VirtualInterface** objects	Specifies parameters for creating a virtual interface.
page_info	**PageInfo** object	Specifies the pagination query information.

Table 5-104 VirtualInterface

Parameter	Type	Description
id	String	Specifies the virtual interface ID. Maximum: **36**
name	String	Specifies the virtual interface name. Maximum: **64**
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**.
bandwidth	Integer	Specifies the virtual interface bandwidth. Minimum: **2** Maximum: **2147483647**
create_time	String	Specifies the time when the virtual interface was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used. Maximum: **255**
description	String	Provides supplementary information about the virtual interface. Maximum: **128**
direct_connect_id	String	Specifies the connection ID. Maximum: **36**

Parameter	Type	Description
service_type	String	Specifies the gateway type, which can be **VGW**, **GDGW**, or **LGW**. Enumeration values: ● **VGW** ● **GDGW** ● **LGW**
status	String	Specifies the operating status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, **PENDING_DELETE**, **DELETED**, **AUTHORIZATION**, or **REJECTED**.
tenant_id	String	Specifies the project ID. Minimum: **32** Maximum: **32**
type	String	Specifies the type of the virtual interface. The value is **private**. Default: **private** Maximum: **255** Enumeration values: ● **private** ● **public**
vgw_id	String	Specifies the virtual gateway ID. Minimum: **36** Maximum: **36**
vlan	Integer	Specifies the VLAN for connecting to the user gateway. The value ranges from **0** to **3999**. Minimum: **0** Maximum: **3999**
route_limit	Integer	Specifies the remote subnet route configurations of the virtual interface. Minimum: **1** Maximum: **200** Default: **50**
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable Bidirectional Forwarding Detection (BFD). The value can be **true** or **false**.

Parameter	Type	Description
lag_id	String	Specifies the ID of the LAG associated with the virtual interface. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual interface belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**
local_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
rate_limit	Boolean	Specifies whether rate limiting is enabled on a virtual interface.
address_family	String	Specifies the address family of the virtual interface, which can be **IPv4** or **IPv6**. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
local_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
lgw_id	String	Specifies the ID of the local gateway, which is used in IES scenarios.

Parameter	Type	Description
gateway_id	String	Specifies the ID of the gateway associated with the virtual interface.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
bgp_route_limit	Integer	Specifies the BGP route configuration.
priority	String	Specifies the priority of a virtual interface. The value can be **normal** or **low**. If the priorities are the same, the virtual interfaces work in load balancing mode. If the priorities are different, the virtual interfaces work in active/standby pairs. Outbound traffic is preferentially forwarded to the normal virtual interface with a higher priority. This option is only supported by virtual interfaces that use BGP routing. Default: **normal** Enumeration values: ● **normal** ● **low**
vif_peers	Array of **VifPeer** objects	Provides information about virtual interface peers.
extend_attribute	**VifExtendAttribute** object	Provides extended parameter information.

Table 5-105 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**

Parameter	Type	Description
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-106 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**

Parameter	Type	Description
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Table 5-107 VifExtendAttribute

Parameter	Type	Description
ha_type	String	Specifies the availability detection type of the virtual interface. Enumeration values: • **nqa** • **bfd**
ha_mode	String	Specifies the availability detection mode. Enumeration values: • **auto_single** • **auto_multi** • **static_single** • **static_multi** • **enhance_nqa**
detect_multiplier	Integer	Specifies the number of detection retries. Default: **5**
min_rx_interval	Integer	Specifies the interval for receiving detection packets. Default: **1000**
min_tx_interval	Integer	Specifies the interval for sending detection packets. Default: **1000**
remote_disclaim	Integer	Specifies the remote identifier of the static BFD session.
local_disclaim	Integer	Specifies the local identifier of the static BFD session.

Table 5-108 PageInfo

Parameter	Type	Description
previous_marker	String	Specifies the marker of the previous page. The value is the resource UUID. Minimum: **0** Maximum: **36**
current_count	Integer	Specifies the number of resources in the current list. Minimum: **0** Maximum: **2000**

Parameter	Type	Description
next_marker	String	Specifies the marker of the next page. The value is the resource UUID. If the value is empty, the resource is on the last page. Minimum: **0** Maximum: **36**

Example Requests

Querying the virtual interface list

```
GET https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-interfaces
```

Example Responses

Status code: 200

OK

```
{
  "virtual_interfaces" : [ {
    "id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
    "name" : "vif-0819",
    "description" : "mytest",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "direct_connect_id" : "4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
    "vgw_id" : "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
    "type" : "private",
    "service_type" : "VGW",
    "vlan" : 332,
    "bandwidth" : 2,
    "status" : "ACTIVE",
    "create_time" : "2022-08-19T11:28:06.000Z",
    "admin_state_up" : true,
    "enable_bfd" : false,
    "route_limit" : 50,
    "enable_nqa" : false,
    "local_gateway_v4_ip" : "1.1.1.1/30",
    "remote_gateway_v4_ip" : "1.1.1.2/30",
    "ies_id" : null,
    "reason" : null,
    "rate_limit" : false,
    "address_family" : "ipv4",
    "local_gateway_v6_ip" : null,
    "remote_gateway_v6_ip" : null,
    "lgw_id" : null,
    "gateway_id" : null,
    "remote_ep_group" : [ "1.1.2.0/30" ],
    "service_ep_group" : [ ],
    "bgp_route_limit" : 100,
    "priority" : "normal",
    "vif_peers" : [ {
      "id" : "c768eb52-12a8-4859-9b43-81194643040c",
      "tenant_id" : "0605768a3300d5762f82c01180692873",
      "name" : "vif-0819",
      "description" : "",
      "address_family" : "ipv4",
      "local_gateway_ip" : "1.1.1.1/30",
      "remote_gateway_ip" : "1.1.1.2/30",
      "route_mode" : "static",
      "bgp_asn" : null,
      "bgp_md5" : null,
      "device_id" : "18.9.215.131",
      "bgp_route_limit" : 100,
      "bgp_status" : null,
      "status" : "ACTIVE",
      "vif_id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
      "receive_route_num" : -1,
      "remote_ep_group" : [ "1.1.2.0/30" ],
      "service_ep_group" : null,
      "enable_bfd" : false,
      "enable_nqa" : false
    } ],
    "enterprise_project_id" : "0"
  } ],
  "request_id" : "5633df7af874576d819a481c76673236"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListVirtualInterfacesSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListVirtualInterfacesRequest request = new ListVirtualInterfacesRequest();
        request.withDirectConnectId();
        request.withVgwId();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortDir();
        request.withSortKey("<sort_key>");
        request.withEnterpriseProjectId();
        request.withId();
        request.withStatus();
        try {
            ListVirtualInterfacesResponse response = client.listVirtualInterfaces(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListVirtualInterfacesRequest()
        request.direct_connect_id = 
        request.vgw_id = 
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields = 
        request.sort_dir = 
        request.sort_key = "<sort_key>"
        request.enterprise_project_id = 
        request.id = 
        request.status = 
        response = client.list_virtual_interfaces(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListVirtualInterfacesRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ListVirtualInterfaces(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.3.5 Creating a Virtual Interface

Function

This API is used to create a virtual interface.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/dcaas/virtual-interfaces

Table 5-109 Path Parameters

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

Request Parameters

Table 5-110 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-111 Request body parameters

Parameter	Mandatory	Type	Description
virtual_interface	Yes	**CreateVirtualInterface** object	Specifies parameters for creating a virtual interface.

Table 5-112 CreateVirtualInterface

Parameter	Mandatory	Type	Description
name	No	String	Specifies the virtual interface name. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the virtual interface. Minimum: **0** Maximum: **128**
direct_connect_id	No	String	Specifies the ID of the connection associated with the virtual interface. When creating a virtual interface, you need to specify **direct_connect_id** or **lag_id**. This parameter is mandatory when LAG is not supported at the site.
type	Yes	String	Specifies the type of the virtual interface. The value is **private**. Enumeration values: ● **private** ● **public**
service_type	No	String	Specifies the gateway type, which can be **VGW**, **GDGW**, or **LGW**. Enumeration values: ● **VGW** ● **GDGW** ● **LGW**
vlan	Yes	Integer	Specifies the customer VLAN to be connected. If you select a hosted connection, the VLAN must be the same as that of the hosted connection. Minimum: **0** Maximum: **3999**

Parameter	Mandatory	Type	Description
bandwidth	Yes	Integer	Specifies the virtual interface bandwidth. Minimum: **2** Maximum: **2147483647**
local_gateway_v4_ip	No	String	Specifies the IPv4 interface address of the gateway used on the cloud. This parameter is mandatory if **address_family** is set to an IPv4 address.
remote_gateway_v4_ip	No	String	Specifies the IPv4 interface address of the gateway on the on-premises network. This parameter is mandatory if **address_family** is set to an IPv4 address.
address_family	No	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**. Default: **ipv4**
local_gateway_v6_ip	No	String	Specifies the IPv6 interface address of the gateway used on the cloud. This parameter is mandatory if **address_family** is set to an IPv6 address.
remote_gateway_v6_ip	No	String	Specifies the IPv6 interface address of the gateway on the on-premises network. This parameter is mandatory if **address_family** is set to an IPv6 address.
vgw_id	Yes	String	Specifies the ID of the virtual gateway connected by the virtual interface.
route_mode	Yes	String	Specifies the routing mode, which can be **static** or **bgp**. Default: **static** Enumeration values: ● **static** ● **bgp**

Parameter	Mandatory	Type	Description
bgp_asn	No	Integer	Specifies the ASN of the BGP peer on the customer side. Minimum: **1** Maximum: **4294967295**
bgp_md5	No	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Yes	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	No	Array of strings	Specifies the subnets that access Internet services through a direct connection.
enable_bfd	No	Boolean	Specifies whether to enable Bidirectional Forwarding Detection (BFD). The value can be **true** or **false**. Default: **false**
enable_nqa	No	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**. Default: **false**
lag_id	No	String	Specifies the ID of the LAG associated with the virtual interface. Minimum: **36** Maximum: **36**
resource_tenant_id	No	String	Specifies the project ID of another tenant, which is used to create virtual interfaces across tenants.
enterprise_project_id	No	String	Specifies the ID of the enterprise project that the resource belongs to. Minimum: **36** Maximum: **36**
tags	No	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**

Table 5-113 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	No	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Response Parameters

Status code: 201

Table 5-114 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
virtual_interface	**VirtualInterface** object	Specifies parameters for creating a virtual interface.

Table 5-115 VirtualInterface

Parameter	Type	Description
id	String	Specifies the virtual interface ID. Maximum: **36**
name	String	Specifies the virtual interface name. Maximum: **64**
admin_state_up	Boolean	Specifies the administrative status, which can be **true** or **false**.

Parameter	Type	Description
bandwidth	Integer	Specifies the virtual interface bandwidth. Minimum: **2** Maximum: **2147483647**
create_time	String	Specifies the time when the virtual interface was created. The UTC time format **yyyy-MM-ddTHH:mm:ss.SSSZ** is used. Maximum: **255**
description	String	Provides supplementary information about the virtual interface. Maximum: **128**
direct_connect_id	String	Specifies the connection ID. Maximum: **36**
service_type	String	Specifies the gateway type, which can be **VGW**, **GDGW**, or **LGW**. Enumeration values: ● **VGW** ● **GDGW** ● **LGW**
status	String	Specifies the operating status, which can be **ACTIVE**, **DOWN**, **BUILD**, **ERROR**, **PENDING_CREATE**, **PENDING_UPDATE**, **PENDING_DELETE**, **DELETED**, **AUTHORIZATION**, or **REJECTED**.
tenant_id	String	Specifies the project ID. Minimum: **32** Maximum: **32**
type	String	Specifies the type of the virtual interface. The value is **private**. Default: **private** Maximum: **255** Enumeration values: ● **private** ● **public**
vgw_id	String	Specifies the virtual gateway ID. Minimum: **36** Maximum: **36**

Parameter	Type	Description
vlan	Integer	Specifies the VLAN for connecting to the user gateway. The value ranges from **0** to **3999**. Minimum: **0** Maximum: **3999**
route_limit	Integer	Specifies the remote subnet route configurations of the virtual interface. Minimum: **1** Maximum: **200** Default: **50**
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable Bidirectional Forwarding Detection (BFD). The value can be **true** or **false**.
lag_id	String	Specifies the ID of the LAG associated with the virtual interface. Minimum: **36** Maximum: **36**
device_id	String	Specifies the ID of the device that the virtual interface belongs to.
enterprise_project_id	String	Specifies the ID of the enterprise project that the virtual interface belongs to. Minimum: **36** Maximum: **36**
tags	Array of **Tag** objects	Specifies the tags. Array Length: **0 - 10**
local_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v4_ip	String	Specifies the IPv4 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
ies_id	String	Specifies the ID of an IES edge site.
reason	String	Displays error information if the status of a line is **Error**.
rate_limit	Boolean	Specifies whether rate limiting is enabled on a virtual interface.

Parameter	Type	Description
address_family	String	Specifies the address family of the virtual interface, which can be **IPv4** or **IPv6**. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
local_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on the cloud. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
remote_gateway_v6_ip	String	Specifies the IPv6 interface address of the gateway used on premises. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
lgw_id	String	Specifies the ID of the local gateway, which is used in IES scenarios.
gateway_id	String	Specifies the ID of the gateway associated with the virtual interface.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections. This parameter has been migrated to the **vifpeer** parameter list and will be discarded later.
bgp_route_limit	Integer	Specifies the BGP route configuration.
priority	String	Specifies the priority of a virtual interface. The value can be **normal** or **low**. If the priorities are the same, the virtual interfaces work in load balancing mode. If the priorities are different, the virtual interfaces work in active/standby pairs. Outbound traffic is preferentially forwarded to the normal virtual interface with a higher priority. This option is only supported by virtual interfaces that use BGP routing. Default: **normal** Enumeration values: ● **normal** ● **low**

Parameter	Type	Description
vif_peers	Array of **VifPeer** objects	Provides information about virtual interface peers.
extend_attribute	**VifExtendAttribute** object	Provides extended parameter information.

Table 5-116 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Table 5-117 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**

Parameter	Type	Description
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**

Parameter	Type	Description
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Table 5-118 VifExtendAttribute

Parameter	Type	Description
ha_type	String	Specifies the availability detection type of the virtual interface. Enumeration values: ● **nqa** ● **bfd**
ha_mode	String	Specifies the availability detection mode. Enumeration values: ● **auto_single** ● **auto_multi** ● **static_single** ● **static_multi** ● **enhance_nqa**
detect_multiplier	Integer	Specifies the number of detection retries. Default: **5**
min_rx_interval	Integer	Specifies the interval for receiving detection packets. Default: **1000**
min_tx_interval	Integer	Specifies the interval for sending detection packets. Default: **1000**
remote_disclaim	Integer	Specifies the remote identifier of the static BFD session.
local_disclaim	Integer	Specifies the local identifier of the static BFD session.

Example Requests

Creating a private virtual interface, with the bandwidth set to 2 Mbit/s, VLAN to 332, local gateway to 1.1.1.1/30, and remote gateway to 1.1.1.2/30, and routing mode to static routing

```
POST https://{dc_endpoint}/v3/0605768a3300d5762f82c01180692873/dcaas/virtual-interfaces

{
  "virtual_interface" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "direct_connect_id" : "4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
    "vgw_id" : "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
    "vlan" : 332,
    "bandwidth" : 2,
    "local_gateway_v4_ip" : "1.1.1.1/30",
    "remote_gateway_v4_ip" : "1.1.1.2/30",
    "type" : "private",
    "route_mode" : "static",
    "remote_ep_group" : [ "1.1.2.0/30" ]
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "virtual_interface" : {
    "id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
    "name" : "vif-0819",
    "description" : "mytest",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "direct_connect_id" : "4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
    "vgw_id" : "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
    "type" : "private",
    "service_type" : "VGW",
    "vlan" : 332,
    "bandwidth" : 2,
    "device_id" : "18.9.215.131",
    "status" : "ACTIVE",
    "create_time" : "2022-08-19T11:28:06.000Z",
    "admin_state_up" : true,
    "enable_bfd" : false,
    "route_limit" : 50,
    "enable_nqa" : false,
    "local_gateway_v4_ip" : "1.1.1.1/30",
    "remote_gateway_v4_ip" : "1.1.1.2/30",
    "ies_id" : null,
    "reason" : null,
    "rate_limit" : false,
    "local_gateway_v6_ip" : null,
    "remote_gateway_v6_ip" : null,
    "lgw_id" : null,
    "lag_id" : null,
    "priority" : "normal",
    "vif_peers" : [ {
      "id" : "c768eb52-12a8-4859-9b43-81194643040c",
      "tenant_id" : "0605768a3300d5762f82c01180692873",
      "name" : "vif-0819",
      "description" : "",
      "address_family" : "ipv4",
      "local_gateway_ip" : "1.1.1.1/30",
      "remote_gateway_ip" : "1.1.1.2/30",
      "route_mode" : "static",
      "bgp_asn" : null,
      "bgp_md5" : null,
      "device_id" : "18.9.215.131",
      "bgp_route_limit" : 100,
      "bgp_status" : null,
      "status" : "ACTIVE",
      "vif_id" : "0d0fdf63-f2c4-491c-8866-d504796189be",
      "receive_route_num" : -1,
      "remote_ep_group" : [ "1.1.2.0/30" ],
      "enable_bfd" : false,
      "enable_nqa" : false,
      "service_ep_group" : null
    } ],
    "enterprise_project_id" : "0"
  },
  "request_id" : "5633df7af874576d819a481c76673236"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a private virtual interface, with the bandwidth set to 2 Mbit/s, VLAN to 332, local gateway to 1.1.1.1/30, and remote gateway to 1.1.1.2/30, and routing mode to static routing

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class CreateVirtualInterfaceSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateVirtualInterfaceRequest request = new CreateVirtualInterfaceRequest();
        CreateVirtualInterfaceRequestBody body = new CreateVirtualInterfaceRequestBody();
        List<String> listVirtualInterfaceRemoteEpGroup = new ArrayList<>();
        listVirtualInterfaceRemoteEpGroup.add("1.1.2.0/30");
        CreateVirtualInterface virtualInterfacebody = new CreateVirtualInterface();
        virtualInterfacebody.withName("vif-0819")
            .withDescription("mytest")
            .withDirectConnectId("4673e339-8412-4ee1-b73e-2ba9cdfa54c1")
            .withType(CreateVirtualInterface.TypeEnum.fromValue("private"))
            .withVlan(332)
            .withBandwidth(2)
            .withLocalGatewayV4Ip("1.1.1.1/30")
            .withRemoteGatewayV4Ip("1.1.1.2/30")
            .withVgwId("8a47064a-f34c-4f94-b7fe-cac456c9b37b")
            .withRouteMode(CreateVirtualInterface.RouteModeEnum.fromValue("static"))
            .withRemoteEpGroup(listVirtualInterfaceRemoteEpGroup);
        body.withVirtualInterface(virtualInterfacebody);
        request.withBody(body);
        try {
            CreateVirtualInterfaceResponse response = client.createVirtualInterface(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Creating a private virtual interface, with the bandwidth set to 2 Mbit/s, VLAN to 332, local gateway to 1.1.1.1/30, and remote gateway to 1.1.1.2/30, and routing mode to static routing

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateVirtualInterfaceRequest()
        listRemoteEpGroupVirtualInterface = [
            "1.1.2.0/30"
        ]
        virtualInterfacebody = CreateVirtualInterface(
            name="vif-0819",
            description="mytest",
            direct_connect_id="4673e339-8412-4ee1-b73e-2ba9cdfa54c1",
            type="private",
            vlan=332,
            bandwidth=2,
            local_gateway_v4_ip="1.1.1.1/30",
            remote_gateway_v4_ip="1.1.1.2/30",
            vgw_id="8a47064a-f34c-4f94-b7fe-cac456c9b37b",
            route_mode="static",
            remote_ep_group=listRemoteEpGroupVirtualInterface
        )
        request.body = CreateVirtualInterfaceRequestBody(
            virtual_interface=virtualInterfacebody
        )
        response = client.create_virtual_interface(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Creating a private virtual interface, with the bandwidth set to 2 Mbit/s, VLAN to 332, local gateway to 1.1.1.1/30, and remote gateway to 1.1.1.2/30, and routing mode to static routing

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateVirtualInterfaceRequest{}
    var listRemoteEpGroupVirtualInterface = []string{
        "1.1.2.0/30",
    }
    nameVirtualInterface:= "vif-0819"
    descriptionVirtualInterface:= "mytest"
    directConnectIdVirtualInterface:= "4673e339-8412-4ee1-b73e-2ba9cdfa54c1"
    localGatewayV4IpVirtualInterface:= "1.1.1.1/30"
    remoteGatewayV4IpVirtualInterface:= "1.1.1.2/30"
    virtualInterfacebody := &model.CreateVirtualInterface{
        Name: &nameVirtualInterface,
        Description: &descriptionVirtualInterface,
        DirectConnectId: &directConnectIdVirtualInterface,
        Type: model.GetCreateVirtualInterfaceTypeEnum().PRIVATE,
        Vlan: int32(332),
        Bandwidth: int32(2),
        LocalGatewayV4Ip: &localGatewayV4IpVirtualInterface,
        RemoteGatewayV4Ip: &remoteGatewayV4IpVirtualInterface,
        VgwId: "8a47064a-f34c-4f94-b7fe-cac456c9b37b",
        RouteMode: model.GetCreateVirtualInterfaceRouteModeEnum().STATIC,
        RemoteEpGroup: listRemoteEpGroupVirtualInterface,
    }
    request.Body = &model.CreateVirtualInterfaceRequestBody{
        VirtualInterface: virtualInterfacebody,
    }
    response, err := client.CreateVirtualInterface(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
201	Created

Error Codes

See **Error Codes**.

5.3.6 Updating a Virtual Interface Peer

Function

This API is used to update a virtual interface peer, for example, its remote subnet, name, and description. This API is available only in regions that support IPv6. To use this API, contact customer service.

Calling Method

For details, see **Calling APIs**.

URI

PUT /v3/{project_id}/dcaas/vif-peers/{vif_peer_id}

Table 5-119 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
vif_peer_id	Yes	String	Specifies the virtual interface peer ID.

Request Parameters

Table 5-120 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-121 Request body parameters

Parameter	Mandatory	Type	Description
vif_peer	No	**UpdateVifPeer** object	Specifies parameters for updating a virtual interface peer.

Table 5-122 UpdateVifPeer

Parameter	Mandatory	Type	Description
name	No	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**

Parameter	Mandatory	Type	Description
description	No	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**
remote_ep_group	No	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.

Response Parameters

Status code: 200

Table 5-123 Response body parameters

Parameter	Type	Description
vif_peer	**VifPeer** object	Specifies the virtual interface peer.

Table 5-124 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**

Parameter	Type	Description
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**

Parameter	Type	Description
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Example Requests

Updating the name, description, and remote subnet of a virtual interface peer

```
PUT https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dcaas/vif-peers/68250543-0a13-4ac7-aa36-d018856ac640

{
  "vif_peer" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "remote_ep_group" : [ "20.1.1.0/24" ]
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "vif_peer" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "id" : "4c95de3e-9f75-4357-9c79-b22498dd71c7",
    "tenant_id" : "ed28c294165741faaeccab26913122a1",
    "address_family" : "ipv4",
    "local_gateway_ip" : "12.3.4.1/30",
    "remote_gateway_ip" : "12.3.4.2/30",
    "route_mode" : "static",
    "bgp_asn" : null,
    "bgp_md5" : null,
    "bgp_route_limit" : 100,
    "bgp_status" : null,
    "status" : "ACTIVE",
    "vif_id" : "5d6c17bc-0ebe-420b-8734-21f519e9d7ad",
    "receive_route_num" : -1,
    "remote_ep_group" : [ "20.1.1.0/24" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Updating the name, description, and remote subnet of a virtual interface peer

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class UpdateVifPeerSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateVifPeerRequest request = new UpdateVifPeerRequest();
        UpdateVifPeerRequestBody body = new UpdateVifPeerRequestBody();
        List<String> listVifPeerRemoteEpGroup = new ArrayList<>();
        listVifPeerRemoteEpGroup.add("20.1.1.0/24");
        UpdateVifPeer vifPeerbody = new UpdateVifPeer();
        vifPeerbody.withName("vif-0819")
            .withDescription("mytest")
            .withRemoteEpGroup(listVifPeerRemoteEpGroup);
        body.withVifPeer(vifPeerbody);
        request.withBody(body);
        try {
            UpdateVifPeerResponse response = client.updateVifPeer(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Updating the name, description, and remote subnet of a virtual interface peer

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateVifPeerRequest()
        listRemoteEpGroupVifPeer = [
            "20.1.1.0/24"
        ]
        vifPeerbody = UpdateVifPeer(
            name="vif-0819",
            description="mytest",
            remote_ep_group=listRemoteEpGroupVifPeer
        )
        request.body = UpdateVifPeerRequestBody(
            vif_peer=vifPeerbody
        )
        response = client.update_vif_peer(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Updating the name, description, and remote subnet of a virtual interface peer

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVifPeerRequest{}
    var listRemoteEpGroupVifPeer = []string{
        "20.1.1.0/24",
    }
    nameVifPeer:= "vif-0819"
    descriptionVifPeer:= "mytest"
    vifPeerbody := &model.UpdateVifPeer{
        Name: &nameVifPeer,
        Description: &descriptionVifPeer,
        RemoteEpGroup: &listRemoteEpGroupVifPeer,
    }
    request.Body = &model.UpdateVifPeerRequestBody{
        VifPeer: vifPeerbody,
    }
    response, err := client.UpdateVifPeer(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.3.7 Deleting a Virtual Interface Peer

Function

This API is used to delete a virtual interface. The virtual interface must contain at least one peer. The last peer cannot be deleted. This API is available only in regions that support IPv6. To use this API, contact customer service.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/dcaas/vif-peers/{vif_peer_id}

Table 5-125 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
vif_peer_id	Yes	String	Specifies the virtual interface peer ID.

Request Parameters

Table 5-126 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

None

Example Requests

Deleting a virtual interface peer

```
DELETE https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dcaas/vif-peers/68250543-0a13-4ac7-aa36-d018856ac640
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteVifPeerSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteVifPeerRequest request = new DeleteVifPeerRequest();
        try {
            DeleteVifPeerResponse response = client.deleteVifPeer(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.3.8 Creating a Virtual Interface Peer

Function

This API is used to create an IPv6 peer. Each virtual interface can have two peers: IPv4 and IPv6 peers. When a virtual interface is created, an IPv4 peer is created by default. After a virtual interface peer is created, you can query its settings by viewing the virtual interface details. This API is available only in regions that support IPv6. To use this API, contact customer service.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/dcaas/vif-peers

Table 5-127 Path Parameters

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

Request Parameters

Table 5-128 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-129 Request body parameters

Parameter	Mandatory	Type	Description
vif_peer	No	**CreateVifPeer** object	Specifies parameters for creating a virtual interface peer.

Table 5-130 CreateVifPeer

Parameter	Mandatory	Type	Description
name	No	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	No	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**
address_family	No	String	Specifies the address family of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	No	String	Specifies the cloud-side gateway IP address of the virtual interface peer.

Parameter	Mandatory	Type	Description
remote_gateway_ip	No	String	Specifies the customer-side gateway address of the virtual interface peer.
route_mode	No	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: - **bgp** - **static**
bgp_asn	No	Integer	Specifies the AS of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	No	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	No	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
vif_id	No	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**

Response Parameters

Status code: 201

Table 5-131 Response body parameters

Parameter	Type	Description
vif_peer	**VifPeer** object	Specifies the virtual interface peer.

Table 5-132 VifPeer

Parameter	Type	Description
id	String	Specifies the resource ID. Minimum: **36** Maximum: **36**
tenant_id	String	Specifies the ID of the project that the virtual interface peer belongs to. Minimum: **36** Maximum: **36**
name	String	Specifies the name of the virtual interface peer. Minimum: **0** Maximum: **64**
description	String	Provides supplementary information about the virtual interface peer. Minimum: **0** Maximum: **128**
address_family	String	Specifies the address family type of the virtual interface, which can be **IPv4** or **IPv6**.
local_gateway_ip	String	Specifies the address of the virtual interface peer used on the cloud.
remote_gateway_ip	String	Specifies the address of the virtual interface peer used in the on-premises data center.
route_mode	String	Specifies the routing mode, which can be **static** or **bgp**. Maximum: **255** Enumeration values: ● **bgp** ● **static**
bgp_asn	Integer	Specifies the ASN of the BGP peer. Minimum: **1** Maximum: **4294967295**
bgp_md5	String	Specifies the MD5 password of the BGP peer.
remote_ep_group	Array of strings	Specifies the remote subnet list, which records the CIDR blocks used in the on-premises data center.
service_ep_group	Array of strings	Specifies the list of public network addresses that can be accessed by the on-premises data center. This field is required in the APIs of public network connections.

Parameter	Type	Description
device_id	String	Specifies the ID of the device that the virtual interface peer belongs to.
bgp_route_limit	Integer	Specifies the BGP route configuration.
bgp_status	String	Specifies the BGP protocol status of the virtual interface peer. If the virtual interface peer uses static routing, the status is **null**. Maximum: **10**
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Specifies the ID of the virtual interface corresponding to the virtual interface peer. Minimum: **36** Maximum: **36**
receive_route_num	Integer	Specifies the number of received BGP routes if BGP routing is used. If static routing is used, this parameter is meaningless and the value is **-1**. Note: If this parameter cannot be obtained, contact customer service to migrate your ports.
enable_nqa	Boolean	Specifies whether to enable NQA. The value can be **true** or **false**.
enable_bfd	Boolean	Specifies whether to enable BFD. The value can be **true** or **false**.

Example Requests

Creating a virtual interface peer with the remote subnet set to 20.1.1.0/24

```
POST https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dcaas/vif-peers/68250543-0a13-4ac7-
aa36-d018856ac640

{
  "vif_peer" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "remote_ep_group" : [ "20.1.1.0/24" ]
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "vif_peer" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "id" : "4c95de3e-9f75-4357-9c79-b22498dd71c7",
    "tenant_id" : "ed28c294165741faaeccab26913122a1",
    "address_family" : "ipv4",
    "local_gateway_ip" : "12.3.4.1/30",
    "remote_gateway_ip" : "12.3.4.2/30",
    "route_mode" : "static",
    "bgp_asn" : null,
    "bgp_md5" : null,
    "bgp_route_limit" : 100,
    "bgp_status" : null,
    "status" : "ACTIVE",
    "vif_id" : "5d6c17bc-0ebe-420b-8734-21f519e9d7ad",
    "receive_route_num" : -1,
    "remote_ep_group" : [ "12.3.4.0/30 172.56.0.0/16" ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a virtual interface peer with the remote subnet set to 20.1.1.0/24

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class CreateVifPeerSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateVifPeerRequest request = new CreateVifPeerRequest();
        CreateVifPeerRequestBody body = new CreateVifPeerRequestBody();
        List<String> listVifPeerRemoteEpGroup = new ArrayList<>();
        listVifPeerRemoteEpGroup.add("20.1.1.0/24");
        CreateVifPeer vifPeerbody = new CreateVifPeer();
        vifPeerbody.withName("vif-0819")
            .withDescription("mytest")
            .withRemoteEpGroup(listVifPeerRemoteEpGroup);
        body.withVifPeer(vifPeerbody);
        request.withBody(body);
        try {
            CreateVifPeerResponse response = client.createVifPeer(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Creating a virtual interface peer with the remote subnet set to 20.1.1.0/24

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateVifPeerRequest()
        listRemoteEpGroupVifPeer = [
            "20.1.1.0/24"
        ]
        vifPeerbody = CreateVifPeer(
            name="vif-0819",
            description="mytest",
            remote_ep_group=listRemoteEpGroupVifPeer
        )
        request.body = CreateVifPeerRequestBody(
            vif_peer=vifPeerbody
        )
        response = client.create_vif_peer(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Creating a virtual interface peer with the remote subnet set to 20.1.1.0/24

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateVifPeerRequest{}
    var listRemoteEpGroupVifPeer = []string{
        "20.1.1.0/24",
    }
    nameVifPeer:= "vif-0819"
    descriptionVifPeer:= "mytest"
    vifPeerbody := &model.CreateVifPeer{
        Name: &nameVifPeer,
        Description: &descriptionVifPeer,
        RemoteEpGroup: &listRemoteEpGroupVifPeer,
    }
    request.Body = &model.CreateVifPeerRequestBody{
        VifPeer: vifPeerbody,
    }
    response, err := client.CreateVifPeer(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
201	Created

Error Codes

See **Error Codes**.

5.3.9 Performing a Virtual Interface Switchover Test

Function

If there are two connections, automatic switchover between the connections is required for connectivity testing.If a switchover is performed, the virtual interface of the connection in use is disabled and traffic is interrupted.You can perform two operations on a virtual interface:- Run the **shutdown** command to disable the virtual interface.- Run the **undo_shutdown** command to enable the virtual interface.If **shutdown** is selected for the switchover test, the virtual interface is in the **ADMIN_SHUTDOWN** state, and no operations can be performed on the virtual interface.If **undo_shutdown** is selected for the switchover test, the virtual interface is in the **ACTIVE** state.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/dcaas/switchover-test

Table 5-133 Path Parameters

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

Request Parameters

Table 5-134 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Table 5-135 Request body parameters

Parameter	Mandatory	Type	Description
switchover_test_record	No	**CreateSwitchoverTest** object	Specifies the parameters for creating a switchover test record.

Table 5-136 CreateSwitchoverTest

Parameter	Mandatory	Type	Description
resource_id	Yes	String	Specifies the ID of the resource on which the switchover test is to be performed.
resource_type	No	String	Specifies the type of the resource on which the switchover test is to be performed. Default: **virtual_interface** Minimum: **0** Maximum: **128** Enumeration values: ● **virtual_interface**

Parameter	Mandatory	Type	Description
operation	Yes	String	Specifies whether to perform a switchover test. Value options: **shutdown** and **undo_shutdown** Minimum: **0** Maximum: **128** Enumeration values: ● **shutdown** ● **undo_shutdown**

Response Parameters

Status code: 201

Table 5-137 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
switchover_test_record	**SwitchoverTestRecord** object	Provides supplementary information about the switchover test record.

Table 5-138 SwitchoverTestRecord

Parameter	Type	Description
id	String	Specifies the unique ID of the switchover test record.
tenant_id	String	Specifies the tenant ID. Minimum: **32** Maximum: **32**
resource_id	String	Specifies the ID of the resource on which the switchover test is to be performed.

Parameter	Type	Description
resource_type	String	Specifies the type of the resource on which the switchover test is to be performed. Default: **virtual_interface** Minimum: **0** Maximum: **128** Enumeration values: ● **virtual_interface**
operation	String	Specifies whether to perform a switchover test. Value options: **shutdown** and **undo_shutdown** Minimum: **0** Maximum: **128** Enumeration values: ● **shutdown** ● **undo_shutdown**
start_time	String	Specifies the start time of the switchover test.
end_time	String	Specifies the end time of the switchover test.
operate_status	String	Specifies the switchover test status. ● **STARTING**: initial status ● **INPROGRESS**: delivering configuration ● **COMPLETE**: configuration delivered ● **ERROR**: Configuration delivery failed Enumeration values: ● **STARTING** ● **INPROGRESS** ● **COMPLETE** ● **ERROR**

Example Requests

Shutting down the virtual interface to perform a switchover test

```
POST https://{dc_endpoint}/v3/de58f033eb664102ba85e4a5db473ca5/dcaas/switchover-test

{
  "switchover_test_record" : {
    "resource_type" : "virtual_interface",
    "resource_id" : "d0b3329c-0063-470c-b1dc-657656b2e540",
    "operation" : "shutdown"
  }
}
```

Example Responses

Status code: 201

Created

```
{
  "switchover_test_record" : {
    "id" : "862d61f1-d9ea-4093-ba0e-2b8d415e3ab3",
    "tenant_id" : "de58f033eb664102ba85e4a5db473ca5",
    "resource_type" : "virtual_interface",
    "resource_id" : "d0b3329c-0063-470c-b1dc-657656b2e540",
    "operation" : "shutdown",
    "start_time" : "2023-10-09T18:41:23.000Z",
    "end_time" : "2023-10-09T18:41:23.000Z",
    "operate_status" : "STARTING"
  },
  "request_id" : "f91634a12b116b6f946d7871f5b4de18"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Shutting down the virtual interface to perform a switchover test

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class SwitchoverTestSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        SwitchoverTestRequest request = new SwitchoverTestRequest();
        CreateSwitchoverTestRequestBody body = new CreateSwitchoverTestRequestBody();
        CreateSwitchoverTest switchoverTestRecordbody = new CreateSwitchoverTest();
        switchoverTestRecordbody.withResourceId("d0b3329c-0063-470c-b1dc-657656b2e540")
            .withResourceType(CreateSwitchoverTest.ResourceTypeEnum.fromValue("virtual_interface"))
            .withOperation(CreateSwitchoverTest.OperationEnum.fromValue("shutdown"));
        body.withSwitchoverTestRecord(switchoverTestRecordbody);
        request.withBody(body);
        try {
            SwitchoverTestResponse response = client.switchoverTest(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Shutting down the virtual interface to perform a switchover test

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = SwitchoverTestRequest()
        switchoverTestRecordbody = CreateSwitchoverTest(
            resource_id="d0b3329c-0063-470c-b1dc-657656b2e540",
            resource_type="virtual_interface",
            operation="shutdown"
        )
        request.body = CreateSwitchoverTestRequestBody(
            switchover_test_record=switchoverTestRecordbody
        )
        response = client.switchover_test(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Shutting down the virtual interface to perform a switchover test

```
package main

```

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SwitchoverTestRequest{}
    resourceTypeSwitchoverTestRecord:=
model.GetCreateSwitchoverTestResourceTypeEnum().VIRTUAL_INTERFACE
    switchoverTestRecordbody := &model.CreateSwitchoverTest{
        ResourceId: "d0b3329c-0063-470c-b1dc-657656b2e540",
        ResourceType: &resourceTypeSwitchoverTestRecord,
        Operation: model.GetCreateSwitchoverTestOperationEnum().SHUTDOWN,
    }
    request.Body = &model.CreateSwitchoverTestRequestBody{
        SwitchoverTestRecord: switchoverTestRecordbody,
    }
    response, err := client.SwitchoverTest(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
201	Created

Error Codes

See **Error Codes**.

5.3.10 Querying the Switchover Test Records of a Virtual Interface

Function

This API is used to query the switchover test record list. Only the records whose **operate_status** is **COMPELTE** are displayed.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/switchover-test

Table 5-139 Path Parameters

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

Table 5-140 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records returned on each page. Value range: 1 to 2000 Minimum: **1** Maximum: **2000** Default: **2000**
marker	No	String	Specifies the ID of the last resource record on the previous page. If this parameter is left blank, the first page is queried. This parameter must be used together with **limit**. Minimum: **0** Maximum: **36**
fields	No	Array	Specifies the list of fields to be displayed. Array Length: **1 - 5**

Parameter	Mandatory	Type	Description
sort_dir	No	Array	Specifies the sorting order of returned results, which can be **asc** (ascending order) or **desc** (descending order). The default value is **asc**.
sort_key	No	String	Specifies the sorting field. Default: **id** Minimum: **0** Maximum: **36**
resource_id	No	Array	Resource ID used for querying switchover test records.

Request Parameters

Table 5-141 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. To obtain the token, see section "Obtaining the User Token" in the *Identity and Access Management API Reference*. The token is the value of **X-Subject-Token** in the response header. Minimum: **0** Maximum: **10240**

Response Parameters

Status code: 200

Table 5-142 Response body parameters

Parameter	Type	Description
request_id	String	Specifies the request ID.
switchover_test_records	Array of **SwitchoverTestRecord** objects	Lists the switchover test records.

Parameter	Type	Description
page_info	**PageInfo** object	Specifies the pagination query information.

Table 5-143 SwitchoverTestRecord

Parameter	Type	Description
id	String	Specifies the unique ID of the switchover test record.
tenant_id	String	Specifies the tenant ID. Minimum: **32** Maximum: **32**
resource_id	String	Specifies the ID of the resource on which the switchover test is to be performed.
resource_type	String	Specifies the type of the resource on which the switchover test is to be performed. Default: **virtual_interface** Minimum: **0** Maximum: **128** Enumeration values: ● **virtual_interface**
operation	String	Specifies whether to perform a switchover test. Value options: **shutdown** and **undo_shutdown** Minimum: **0** Maximum: **128** Enumeration values: ● **shutdown** ● **undo_shutdown**
start_time	String	Specifies the start time of the switchover test.
end_time	String	Specifies the end time of the switchover test.

Parameter	Type	Description
operate_status	String	Specifies the switchover test status. - **STARTING**: initial status - **INPROGRESS**: delivering configuration - **COMPLETE**: configuration delivered - **ERROR**: Configuration delivery failed Enumeration values: - **STARTING** - **INPROGRESS** - **COMPLETE** - **ERROR**

Table 5-144 PageInfo

Parameter	Type	Description
previous_marker	String	Specifies the marker of the previous page. The value is the resource UUID. Minimum: **0** Maximum: **36**
current_count	Integer	Specifies the number of resources in the current list. Minimum: **0** Maximum: **2000**
next_marker	String	Specifies the marker of the next page. The value is the resource UUID. If the value is empty, the resource is on the last page. Minimum: **0** Maximum: **36**

Example Requests

Querying the switchover test record list

```
GET https://{dc_endpoint}/v3/de58f033eb664102ba85e4a5db473ca5/dcaas/switchover-test
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "bb154519fb167f99cdb01b7b9995ad14",
  "switchover_test_records" : [ {
    "id" : "862d61f1-d9ea-4093-ba0e-2b8d415e3ab3",
    "tenant_id" : "de58f033eb664102ba85e4a5db473ca5",
    "resource_type" : "virtual_interface",
    "resource_id" : "d0b3329c-0063-470c-b1dc-657656b2e540",
    "operation" : "shutdown",
    "start_time" : "2023-10-10T10:41:23.000Z",
    "end_time" : "2023-10-10T10:41:23.000Z",
    "operate_status" : "COMPLETE"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListSwitchoverTestRecordsSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListSwitchoverTestRecordsRequest request = new ListSwitchoverTestRecordsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withFields();
        request.withSortDir();
        request.withSortKey("<sort_key>");
        request.withResourceId();
        try {
            ListSwitchoverTestRecordsResponse response = client.listSwitchoverTestRecords(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListSwitchoverTestRecordsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.fields = 
        request.sort_dir = 
        request.sort_key = "<sort_key>"
        request.resource_id = 
        response = client.list_switchover_test_records(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSwitchoverTestRecordsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    sortKeyRequest:= "<sort_key>"
    request.SortKey = &sortKeyRequest
    response, err := client.ListSwitchoverTestRecords(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.4 Tag management

5.4.1 Querying Tags by Resource Type

Function

This API is used to query the tags of a resource type in a specific project. TMS uses this API to list tags created by a tenant to facilitate tag creation and resource filtering on the console.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/{resource_type}/tags

Table 5-145 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_type	Yes	String	Specifies the Direct Connect resource type. ● **dc-directconnect**: connection ● **dc-vgw**: virtual gateway ● **dc-vif**: virtual interface Enumeration values: ● **dc-directconnect** ● **dc-vgw** ● **dc-vif**

Request Parameters

None

Response Parameters

Status code: 200

Table 5-146 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	Specifies the tags.
request_id	String	Specifies the request ID.

Table 5-147 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**

Parameter	Type	Description
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

```
GET https://{dc_endpoint}/v3/ed28c294165741faaeccab26913122a1/dc-directconnect/tags
```

Example Responses

Status code: 200

OK

```
{
  "tags" : [ {
    "key" : "department",
    "value" : "finance"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class ListProjectTagsSolution {

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

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

        DcClient client = DcClient.newBuilder()
```

```
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListProjectTagsRequest request = new ListProjectTagsRequest();
        try {
            ListProjectTagsResponse response = client.listProjectTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.4.2 Querying Resource Tags

Function

This API is used to query resource tags.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/{resource_type}/{resource_id}/tags

Table 5-148 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_type	Yes	String	Specifies the Direct Connect resource type. ● **dc-directconnect**: connection ● **dc-vgw**: virtual gateway ● **dc-vif**: virtual interface Enumeration values: ● **dc-directconnect** ● **dc-vgw** ● **dc-vif**
resource_id	Yes	String	Specifies the resource ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-149 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	Specifies the tag list.
sys_tags	Array of **Tag** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
request_id	String	Specifies the request ID.

Table 5-150 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**

Parameter	Type	Description
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

Querying tags of a resource

```
https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/resource-instances/action
```

Example Responses

Status code: 200

OK

```
{
  "request_id" : "80ef5f21-b81a-4546-b23d-84272507d330",
  "tags" : [ {
    "key" : "key2",
    "value" : "value2"
  }, {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key3",
    "value" : "value3"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class ShowResourceTagSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowResourceTagRequest request = new ShowResourceTagRequest();
        try {
            ShowResourceTagResponse response = client.showResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main
```

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.4.3 Adding a Resource Tag

Function

- A resource can have a maximum of 10 tags.
- This API is idempotent.

- If a to-be-created tag has the same key as an existing tag, the tag will be created and overwrite the existing one.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/{resource_type}/{resource_id}/tags

Table 5-151 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_id	Yes	String	Specifies the resource ID.
resource_type	Yes	String	Specifies the Direct Connect resource type. ● **dc-directconnect**: connection ● **dc-vgw**: virtual gateway ● **dc-vif**: virtual interface Enumeration values: ● **dc-directconnect** ● **dc-vgw** ● **dc-vif**

Request Parameters

Table 5-152 Request body parameters

Parameter	Mandatory	Type	Description
tag	Yes	**Tag** object	Specifies the resource tag.

Table 5-153 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	No	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Response Parameters

None

Example Requests

Adding tags to a resource

```
POST https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/3320166e-b937-40cc-
a35c-02cd3f2b3ee2/tags

{
  "tag" : {
    "key" : "key1",
    "value" : "value1"
  }
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Adding tags to a resource

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class CreateResourceTagSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateResourceTagRequest request = new CreateResourceTagRequest();
        CreateResourceTagRequestBody body = new CreateResourceTagRequestBody();
        Tag tagbody = new Tag();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateResourceTagResponse response = client.createResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Adding tags to a resource

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateResourceTagRequest()
        tagbody = Tag(
            key="key1",
            value="value1"
        )
        request.body = CreateResourceTagRequestBody(
            tag=tagbody
        )
        response = client.create_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Adding tags to a resource

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateResourceTagRequest{}
    valueTag:= "value1"
    tagbody := &model.Tag{
        Key: "key1",
        Value: &valueTag,
    }
    request.Body = &model.CreateResourceTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.4.4 Adding or Deleting Tags in Batches

Function

This API is used to batch add tags to or delete tags from a specified resource. TMS needs to use this API to manage tags of resources in batches. A resource can have a maximum of 10 tags.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/{resource_type}/{resource_id}/tags/action

Table 5-154 Path Parameters

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

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Specifies the Direct Connect resource type. - **dc-directconnect**: connection - **dc-vgw**: virtual gateway - **dc-vif**: virtual interface Enumeration values: - **dc-directconnect** - **dc-vgw** - **dc-vif**

Request Parameters

Table 5-155 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Specifies the operation. The options are as follows: - **create** - **delete** Enumeration values: - **create** - **delete**
tags	No	Array of Tag objects	Specifies the tags.
sys_tags	No	Array of Tag objects	Specifies the system tags.

Table 5-156 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**

Parameter	Mandatory	Type	Description
value	No	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Response Parameters

None

Example Requests

- Batch adding tags to resources

```
POST https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/3320166e-b937-40cc-
a35c-02cd3f2b3ee2/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

- Batch deleting resource tags

```
POST https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/3320166e-b937-40cc-
a35c-02cd3f2b3ee2/tags/action

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

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

- Batch adding tags to resources

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

```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class BatchCreateResourceTagsSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateResourceTagsRequest request = new BatchCreateResourceTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateResourceTagsResponse response = client.batchCreateResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Batch deleting resource tags

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class BatchCreateResourceTagsSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateResourceTagsRequest request = new BatchCreateResourceTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchCreateResourceTagsResponse response = client.batchCreateResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Batch adding tags to resources

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateResourceTagsRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Batch deleting resource tags

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateResourceTagsRequest()
        listTagsbody = [
            Tag(
                key="key1"
            ),
            Tag(
                key="key2",
                value="value3"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="delete"
        )
        response = client.batch_create_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Batch adding tags to resources

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagsRequest{}
    valueTags:= "value1"
    valueTags1:= "value2"
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: &valueTags,
        },
        {
            Key: "key2",
            Value: &valueTags1,
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: &listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().CREATE,
    }
    response, err := client.BatchCreateResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Batch deleting resource tags

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagsRequest{}
    valueTags:= "value3"
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
        },
        {
            Key: "key2",
            Value: &valueTags,
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: &listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().DELETE,
    }
    response, err := client.BatchCreateResourceTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.4.5 Deleting a Resource Tag

Function

When a tag is deleted, the tag character set is not verified. Before calling this API, the API URL must be encoded. If the key to be deleted does not exist, 404 is displayed. The key cannot be left blank or an empty string.

Calling Method

For details, see **Calling APIs**.

URI

DELETE /v3/{project_id}/{resource_type}/{resource_id}/tags/{key}

Table 5-157 Path Parameters

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key.
project_id	Yes	String	Specifies the project ID.
resource_id	Yes	String	Specifies the resource ID.

Parameter	Mandatory	Type	Description
resource_type	Yes	String	Specifies the Direct Connect resource type. - **dc-directconnect**: connection - **dc-vgw**: virtual gateway - **dc-vif**: virtual interface Enumeration values: - **dc-directconnect** - **dc-vgw** - **dc-vif**

Request Parameters

None

Response Parameters

None

Example Requests

Deleting tags from a resource

```
DELETE https://{dc-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/key1
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;


public class DeleteResourceTagSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteResourceTagRequest request = new DeleteResourceTagRequest();
        try {
            DeleteResourceTagResponse response = client.deleteResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
204	No Content

Error Codes

See **Error Codes**.

5.4.6 Querying Resources by Tag

Function

This API is used to query resources such as virtual gateways and virtual interfaces by tag.

Calling Method

For details, see **Calling APIs**.

URI

POST /v3/{project_id}/{resource_type}/resource-instances/action

Table 5-158 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
resource_type	Yes	String	Specifies the Direct Connect resource type. ● **dc-directconnect**: connection ● **dc-vgw**: virtual gateway ● **dc-vif**: virtual interface Enumeration values: ● **dc-directconnect** ● **dc-vgw** ● **dc-vif**

Request Parameters

Table 5-159 Request body parameters

Parameter	Mandatory	Type	Description
offset	No	String	Specifies the index position. The query starts from the next data record indexed by this parameter. You do not need to specify this parameter when you query resources on the first page. When you query resources on subsequent pages, set this parameter to the value returned in the response body for the previous query. This parameter is not available when **action** is set to **count**. If **action** is set to **filter**, the value must be a number, and the default value is **0**. The value cannot be a negative number.
limit	No	String	Specifies the number of records to be queried. This parameter is not available when **action** is set to **count**. If **action** is set to **filter**, the default value is **1000**. The maximum value is **1000**, and the minimum value is **1**. The value cannot be a negative number.
action	Yes	String	Specifies the operation to perform, which can only be **filter** (filtering) or **count** (querying the total number). **filter** indicates pagination query. **count** indicates that the total number of query results meeting the search criteria will be returned. Returning other fields is not allowed. Enumeration values: ● **filter** ● **count**

Parameter	Mandatory	Type	Description
matches	No	Array of **Match** objects	Specifies the search criteria. **key** is the field to match, for example, **resource_name**. **value** indicates the value to be matched. This field is a fixed dictionary value.Determine whether fuzzy match is required based on different fields. For example, if **key** is **resource_name**, fuzzy search (case insensitive) is used by default. If **value** is an empty string, exact match is used. If **key** is **resource_id**, exact match is used. Currently, only **resource_name** is supported.
not_tags	No	Array of **Tags** objects	Specifies tags that are not included. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Keys must be unique and values of a key must be unique. Resources not identified by different keys are in AND relationship, and values in one tag are in OR relationship. If no filtering condition is specified, full data is returned.
tags	No	Array of **Tags** objects	Specifies tags that are included. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Keys must be unique and values of a key must be unique. Resources identified by different keys are in AND relationship, and values in one tag are in OR relationship. If no tag filtering criteria is specified, full data is returned.

Parameter	Mandatory	Type	Description
tags_any	No	Array of **Tags** objects	Specifies any tag that is included. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Keys must be unique and values of a key must be unique. Resources identified by different keys are in OR relationship, and values in one tag are in OR relationship. If no filtering condition is specified, full data is returned.
not_tags_any	No	Array of **Tags** objects	Specifies any tag that is not included. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Keys must be unique and values of a key must be unique. Resources not identified by different keys are in OR relationship, and values in one tag are in OR relationship. If no filtering condition is specified, full data is returned.

Parameter	Mandatory	Type	Description
sys_tags	No	Array of **Tags** objects	Only users with the **op_service** permission can use this field to filter resources. Only one tag structure is contained when this API is called by Tag Management Service (TMS). **key** is **_sys_enterprise_project_id**, and **value** is the enterprise project ID list. When TMS invokes this API, a key can have only one value. **0** indicates the default enterprise project. **sys_tags** and tenant tag filtering conditions (**without_any_tag**, **tags**, **tags_any**, **not_tags**, and **not_tags_any**) cannot be used at the same time.

Table 5-160 Match

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The value can be **dc-directconnect**, **dc-vgw**, or **dc-vif**.
value	Yes	String	Specifies the tag value. Each value can contain a maximum of 255 Unicode characters.

Table 5-161 Tags

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. Each tag key can contain a maximum of 127 Unicode characters. **key** must be specified.
values	Yes	Array of strings	Specifies the list of tag values. Each value can contain a maximum of 255 Unicode characters.

Response Parameters

Status code: 200

Table 5-162 Response body parameters

Parameter	Type	Description
resources	Array of **Resource** objects	Specifies the resource list.
total_count	Integer	Specifies the total number of records.
request_id	String	Specifies the request ID.

Table 5-163 Resource

Parameter	Type	Description
resource_detail	Object	Specifies the resource details.
resource_id	String	Specifies the resource ID.
resource_name	String	Specifies the resource name. This parameter is an empty string by default if there is no resource name.
tags	Array of **Tag** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.
sys_tags	Array of **Tag** objects	Specifies the list of queried tags. If no tag is matched, an empty array is returned.

Table 5-164 Tag

Parameter	Type	Description
key	String	Specifies the tag key. The key can contain a maximum of 36 Unicode characters, including letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **36**
value	String	Specifies the tag value. The value can contain a maximum of 43 Unicode characters, including letters, digits, hyphens (-), underscores (_), and periods (.). Minimum: **0** Maximum: **43**

Example Requests

- Querying resources (**action** set to **filter**)

```
https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/resource-instances/action

{
  "offset" : "10",
  "limit" : "10",
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ]
}
```

- Querying the total number of resources (**action** set to **count**)

```
https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/resource_instances/action

{
  "action" : "count",
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "*value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "sys_tags" : [ {
    "key" : "_sys_enterprise_project_id",
    "values" : [ "5aa119a8-d25b-45a7-8d1b-88e127885635" ]
  } ],
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ]
}
```

Example Responses

Status code: 200

OK

```
{
  "resources" : [ ],
  "total_count" : 0,
  "request_id" : "9a4f4dfc4fb2fc101e65bba07d908535"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Querying resources (**action** set to **filter**)

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ListTagResourceInstancesSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListTagResourceInstancesRequest request = new ListTagResourceInstancesRequest();
        ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<Tags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<Tags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value1");
        listTagsValues.add("value2");
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value1");
        listNotTagsValues.add("value2");
        List<Tags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<Match> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new Match()
                .withKey("resource_name")
                .withValue("resource1")
        );
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("filter"));
        body.withLimit("10");
        body.withOffset("10");
        request.withBody(body);
        try {
            ListTagResourceInstancesResponse response = client.listTagResourceInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Querying the total number of resources (**action** set to **count**)

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

```
public class ListTagResourceInstancesSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ListTagResourceInstancesRequest request = new ListTagResourceInstancesRequest();
        ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
        List<String> listSysTagsValues = new ArrayList<>();
        listSysTagsValues.add("5aa119a8-d25b-45a7-8d1b-88e127885635");
        List<Tags> listbodySysTags = new ArrayList<>();
        listbodySysTags.add(
            new Tags()
                .withKey("_sys_enterprise_project_id")
                .withValues(listSysTagsValues)
        );
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<Tags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<Tags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value1");
        listTagsValues.add("value2");
        List<String> listTagsValues1 = new ArrayList<>();
        listTagsValues1.add("value1");
        listTagsValues1.add("value2");
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsValues1)
        );
        listbodyTags.add(
            new Tags()
                .withKey("key2")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("value1");
        listNotTagsValues.add("*value2");
        List<Tags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<Match> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new Match()
                .withKey("resource_name")
                .withValue("resource1")
        );
        body.withSysTags(listbodySysTags);
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("count"));
        request.withBody(body);
        try {
            ListTagResourceInstancesResponse response = client.listTagResourceInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Querying resources (**action** set to **filter**)

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTagResourceInstancesRequest()
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value1",
            "value2"
        ]
        listTagsbody = [
            Tags(
                key="key1",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            Tags(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            Match(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ListTagResourceInstancesRequestBody(
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody,
            action="filter",
            limit="10",
            offset="10"
        )
        response = client.list_tag_resource_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Querying the total number of resources (**action** set to **count**)

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTagResourceInstancesRequest()
        listValuesSysTags = [
            "5aa119a8-d25b-45a7-8d1b-88e127885635"
        ]
        listSysTagsbody = [
            Tags(
                key="_sys_enterprise_project_id",
                values=listValuesSysTags
            )
        ]
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listValuesTags1 = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            Tags(
                key="key1",
                values=listValuesTags1
            ),
            Tags(
                key="key2",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "value1",
            "*value2"
        ]
        listNotTagsbody = [
            Tags(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            Match(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ListTagResourceInstancesRequestBody(
            sys_tags=listSysTagsbody,
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody,
            action="count"
        )
        response = client.list_tag_resource_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Querying resources (**action** set to **filter**)

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTagResourceInstancesRequest{}
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
    var listTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listMatchesbody = []model.Match{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    limitListTagResourceInstancesRequestBody:= "10"
    offsetListTagResourceInstancesRequestBody:= "10"
    request.Body = &model.ListTagResourceInstancesRequestBody{
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
        Action: model.GetListTagResourceInstancesRequestBodyActionEnum().FILTER,
        Limit: &limitListTagResourceInstancesRequestBody,
        Offset: &offsetListTagResourceInstancesRequestBody,
    }
    response, err := client.ListTagResourceInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Querying the total number of resources (**action** set to **count**)

```
package main

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

```
)

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTagResourceInstancesRequest{}
    var listValuesSysTags = []string{
        "5aa119a8-d25b-45a7-8d1b-88e127885635",
    }
    var listSysTagsbody = []model.Tags{
        {
            Key: "_sys_enterprise_project_id",
            Values: listValuesSysTags,
        },
    }
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    var listValuesTags1 = []string{
        "value1",
        "value2",
    }
    var listTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTags1,
        },
        {
            Key: "key2",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "value1",
        "*value2",
    }
    var listNotTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listMatchesbody = []model.Match{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    request.Body = &model.ListTagResourceInstancesRequestBody{
        SysTags: &listSysTagsbody,
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
        Action: model.GetListTagResourceInstancesRequestBodyActionEnum().COUNT,
    }
    response, err := client.ListTagResourceInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

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

Status Codes

Status Code	Description
200	OK

Error Codes

See **Error Codes**.

5.5 Quota Management

5.5.1 Querying Resource Quotas

Function

This API is used to query the usage of resources, for example, how many connections and virtual interfaces have been created.

Calling Method

For details, see **Calling APIs**.

URI

GET /v3/{project_id}/dcaas/quotas

Table 5-165 Path Parameters

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

Table 5-166 Query Parameters

Parameter	Mandatory	Type	Description
type	No	Array	Specifies the resource quotas. ● **physicalConnect**: quota and usage of the connection ● **virtualInterface**: quota and usage of the virtual interface ● **globalDcGateway**: quota and usage of the global DC gateway ● **peerLinkPerGdgw**: quota and usage of the peer links established with a global DC gateway

Request Parameters

None

Response Parameters

Status code: 201

Table 5-167 Response body parameters

Parameter	Type	Description
quotas	**quotas** object	Provides used quota details.

Table 5-168 quotas

Parameter	Type	Description
resources	Array of **Info** objects	Lists the used quotas of resources.

Table 5-169 Info

Parameter	Type	Description
type	String	Specifies the quota type.
quota	Long	Specifies the available quota. The value **-1** indicates that there is no quota limit.
used	Long	Specifies the used quotas.
unit	String	Specifies the measurement unit of resource usage.

Example Requests

Querying quotas

```
GET https://{dc_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/dcaas/quotas
```

Example Responses

Status code: 201

Quotas are queried.

```
{
  "quotas" : {
    "resources" : [ {
      "type" : "direct_connnect",
      "quota" : 1,
      "used" : 0,
      "unit" : "count"
    } ]
  }
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.dc.v3.region.DcRegion;
import com.huaweicloud.sdk.dc.v3.*;
import com.huaweicloud.sdk.dc.v3.model.*;

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

public class ShowQuotasSolution {

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

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

        DcClient client = DcClient.newBuilder()
                .withCredential(auth)
                .withRegion(DcRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowQuotasRequest request = new ShowQuotasRequest();
        request.withType();
        try {
            ShowQuotasResponse response = client.showQuotas(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdc.v3 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

    client := dc.NewDcClient(
        dc.DcClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

More

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

Status Codes

Status Code	Description
201	Quotas are queried.

Error Codes

See **Error Codes**.

6 Public Parameters

6.1 Common Status Codes

Successful Response	Message	Description
200	OK	Normal response code for GET and PUT operations
201	Created	Normal response code for POST operations
204	No Content	Normal response code for DELETE operations

Error Response	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter the 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.
406 Not Acceptable	The response generated by the server could not be received by the client.
407 Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.

Error Response	Description
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	The request is not completed because the service is abnormal.
501 Not Implemented	The request is not completed because the server does not support the requested function.
502 Bad Gateway	The request is not completed because the server receives an invalid response from an upstream server.
503 Service Unavailable	The request is not completed because the system is out of service temporarily.
504 Gateway Timeout	A gateway timed out.

6.2 Error Codes

Status Code	Error Code	Description	Solution
400	DC.0000	The request body is abnormal.	Correct the parameter settings and send the request again.
400	DC.0001	Request parameter error.	Correct the parameter settings and send the request again.
400	DC.0002	The resource does not exist.	Correct the parameter settings and send the request again.
400	DC.0003	System error.	Correct the parameter settings and send the request again.
400	DC.0004	Invalid IP address.	Correct the parameter settings and send the request again.
400	DC.0005	Failed to call the VPC API.	Correct the parameter settings and send the request again.

Status Code	Error Code	Description	Solution
400	DC.0006	Failed to call the MO API.	Correct the parameter settings and send the request again.
400	DC.0007	The VPC does not exist.	Correct the parameter settings and send the request again.
400	DC.0008	The BGP ASN is out of range.	Correct the parameter settings and send the request again.
400	DC.0009	Invalid policy.	Contact technical support.
400	DC.0010	The VPC_VNI does not exist.	Contact technical support.
400	DC.0011	The AGENT is abnormal.	Correct the parameter settings and send the request again.
400	DC.0012	Failed to call the CBC API.	Contact technical support.
400	DC.0013	You have no permission to operate the field.	Contact technical support.
400	DC.1000	Insufficient connection bandwidth.	Contact technical support.
400	DC.1001	A VLAN is specified for a non-hosted connection.	Contact technical support.
400	DC.1002	Failed to create a hosted connection.	Contact technical support.
400	DC.1003	A VLAN is not specified for a hosted connection.	Contact technical support.
400	DC.1004	A non-operations connection is selected for creating a hosted connection.	Contact technical support.
400	DC.1005	The connection is abnormal.	Contact technical support.
400	DC.1006	The device and type cannot be modified when the connection is in use.	Contact technical support.
400	DC.1007	The connection is in use.	Contact technical support.

Status Code	Error Code	Description	Solution
400	DC.1008	Hosted connection VLANs conflict.	Contact technical support.
400	DC.1010	Connection bandwidth is decreased.	Contact technical support.
400	DC.1011	Connection bandwidth cannot be changed.	Contact technical support.
400	DC.1012	The connection does not exist.	Contact technical support.
400	DC.1014	A redundant connection is set for a non-standard connection.	Contact technical support.
400	DC.1015	The connection or LAG does not exist.	Contact technical support.
400	DC.1017	The VLAN ID of the hosted connection is 0.	Contact technical support.
400	DC.1019	The connection quota has been used up.	Contact technical support.
400	DC.1100	The local endpoint group ID of the virtual gateway is invalid.	Contact technical support.
400	DC.1101	The number of local subnets exceeds the limit.	Contact technical support.
400	DC.1102	The standby device is specified when the active device is not specified.	Contact technical support.
400	DC.1103	The active and standby devices are the same.	Contact technical support.
400	DC.1104	The device information cannot be updated for the virtual gateway.	Contact technical support.
400	DC.1105	The local CIDR block overlap with the remote CIDR block.	Contact technical support.
400	DC.1106	The virtual gateway is in use.	Contact technical support.
400	DC.1107	Virtual gateway VLANs are exhausted.	Contact technical support.
400	DC.1108	Virtual gateway VNIs are exhausted.	Contact technical support.

Status Code	Error Code	Description	Solution
400	DC.1109	Virtual gateway VRFs are exhausted.	Contact technical support.
400	DC.1110	A virtual gateway has been associated with the VPC.	Contact technical support.
400	DC.1111	The virtual gateway does not exist.	Contact technical support.
400	DC.1112	The devices used by the virtual gateway are not in active/standby mode.	Contact technical support.
400	DC.1113	The virtual gateway is in the cutover state.	Contact technical support.
400	DC.1114	**traffic_mode** cannot be updated for VLANs or virtual gateways with a single VTEP device.	Contact technical support.
400	DC.1115	The VNI of the virtual gateway does not exist.	Contact technical support.
400	DC.1116	Failed to update the route gateway.	Contact technical support.
400	DC.1117	The virtual gateway has been associated with two connections and cannot be associated with more connections.	Contact technical support.
400	DC.1118	The virtual gateway is being operated (locked).	Contact technical support.
400	DC.1119	Failed to create the route gateway device group.	Contact technical support.
400	DC.1120	The **main_az_list** configuration item is incorrect.	Contact technical support.
400	DC.1200	No LAG or connection is specified for the virtual interface.	Contact technical support.
400	DC.1201	The connection does not match the virtual gateway type.	Contact technical support.
400	DC.1202	Virtual interfaces of the **double ipsec** type cannot be created.	Contact technical support.

Status Code	Error Code	Description	Solution
400	DC.1203	No ASN is specified for the BGP virtual interface.	Contact technical support.
400	DC.1204	Creating virtual interfaces is not supported for operations connections.	Contact technical support.
400	DC.1205	The status of the resource associated with the virtual interface is abnormal.	Contact technical support.
400	DC.1206	The hosted connection has been associated with a virtual interface.	Contact technical support.
400	DC.1207	The VLAN of the virtual interface is inconsistent with that of the hosted connection.	Contact technical support.
400	DC.1208	The bandwidth of the virtual interface associated with a hosted connection cannot be modified.	Contact technical support.
400	DC.1209	Virtual interface VLANs conflict.	Contact technical support.
400	DC.1210	The virtual interface status cannot be changed.	Contact technical support.
400	DC.1211	The virtual interface does not exist.	Contact technical support.
400	DC.1212	The virtual gateway is in a different group from the connection.	Contact technical support.
400	DC.1213	The virtual interface does not match the tenant.	Contact technical support.
400	DC.1214	The virtual interface has been used.	Contact technical support.
400	DC.1215	The virtual interface type does not match.	Contact technical support.
400	DC.1216	The local gateway is invalid.	Contact technical support.
400	DC.1217	The IES edge site ID of the connection does not match that of the local gateway.	Contact technical support.

Status Code	Error Code	Description	Solution
400	DC.1218	The local gateway does not match.	Contact technical support.
400	DC.1219	A virtual interface with VLAN 0 is created.	Contact technical support.
400	DC.1220	The **vif email** field on the public network is empty.	Contact technical support.
400	DC.1221	The selected virtual gateway device does not support access from Direct Connect locations.	Contact technical support.
400	DC.1222	Insufficient POP VNIs.	Contact technical support.
400	DC.1223	The local BGP ASN is the same as the remote BGP ASN.	Contact technical support.
400	DC.1224	The virtual interface of the connection terminated at a Direct Connect location does not support VLAN 0.	Contact technical support.
400	DC.1225	The number of virtual interfaces for which traffic statistics collection is enabled reaches the limit.	Contact technical support.
400	DC.1226	Traffic statistics collection cannot be enabled on Layer 2 remote interfaces that have no sub-interfaces.	Contact technical support.
400	DC.1227	Traffic statistics cannot be updated because the virtual interface is abnormal.	Contact technical support.
400	DC.1400	The number of endpoint groups reaches the limit.	Contact technical support.
400	DC.1401	Duplicate CIDR blocks.	Contact technical support.
400	DC.1402	The endpoint group is in use.	Contact technical support.
400	DC.1403	The endpoint group does not exist.	Contact technical support.

A Appendixes

A.1 Obtaining a Project ID

Scenarios

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

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

Obtain the Project ID by Calling an API

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

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

The following is an example response. The value of **id** is the project ID.

```
{
    "projects": [
        {
            "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
            "is_domain": false,
            "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
            },
            "id": "a4adasfjljaaaakla12334jklga9sasfg",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

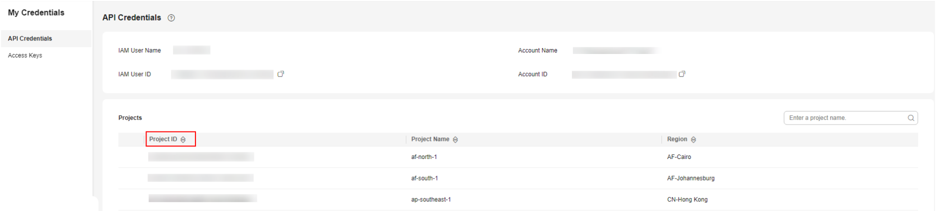
Obtain a Project ID from the Console

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

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

   On the **API Credentials** page, view the project ID in the project list.

Figure A-1 Viewing the project ID



IAM User Name [redacted]

Account Name [redacted]

IAM User ID [redacted]

Account ID [redacted]

Projects

Project ID	Project Name	Region
[redacted]	af-north-1	AF-Cairo
[redacted]	af-south-1	AF-Johannesburg
[redacted]	ap-southeast-1	CN-Hong Kong