

Direct Connect

Referencia de las API

Edición 01
Fecha 2024-09-14



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. Todos los derechos reservados.

Quedan terminantemente prohibidas la reproducción y/o la divulgación totales y/o parciales del presente documento de cualquier forma y/o por cualquier medio sin la previa autorización por escrito de Huawei Cloud Computing Technologies Co., Ltd.

Marcas registradas y permisos



El logotipo HUAWEI y otras marcas registradas de Huawei pertenecen a Huawei Technologies Co., Ltd. Todas las demás marcas registradas y los otros nombres comerciales mencionados en este documento son propiedad de sus respectivos titulares.

Aviso

Es posible que la totalidad o parte de los productos, las funcionalidades y/o los servicios que figuran en el presente documento no se encuentren dentro del alcance de un contrato vigente entre Huawei Cloud y el cliente. Las funcionalidades, los productos y los servicios adquiridos se limitan a los estipulados en el respectivo contrato. A menos que un contrato especifique lo contrario, ninguna de las afirmaciones, informaciones ni recomendaciones contenidas en el presente documento constituye garantía alguna, ni expresa ni implícita.

Huawei está permanentemente preocupada por la calidad de los contenidos de este documento; sin embargo, ninguna declaración, información ni recomendación aquí contenida constituye garantía alguna, ni expresa ni implícita. La información contenida en este documento se encuentra sujeta a cambios sin previo aviso.

Huawei Cloud Computing Technologies Co., Ltd.

Dirección: Huawei Cloud Data Center Jiaoxinggong Road
Avenida Qianzhong
Nuevo distrito de Gui'an
Gui Zhou, 550029
República Popular China

Sitio web: <https://www.huaweicloud.com/intl/es-us/>

Índice

1 Antes de comenzar.....	1
1.1 Introducción.....	1
1.2 Invocación a API.....	1
1.3 Puntos de conexión.....	1
1.4 Restricciones.....	1
2 Descripción de las API.....	3
3 Llamada a APIs.....	4
3.1 Hacer una solicitud de API.....	4
3.2 Autenticación.....	8
4 Uso de API.....	11
5 Las API.....	12
5.1 Conexión.....	12
5.1.1 Consulta de detalles acerca de una conexión.....	12
5.1.2 Actualización de una conexión.....	23
5.1.3 Eliminación de una conexión.....	34
5.1.4 Consulta de la lista de conexiones.....	38
5.1.5 Consulta de la lista de conexiones alojadas.....	49
5.1.6 Creación de una conexión alojada.....	58
5.1.7 Consulta de detalles sobre una conexión alojada.....	67
5.1.8 Actualización de una conexión alojada.....	75
5.1.9 Eliminación de una conexión alojada.....	83
5.2 Gateway virtual.....	87
5.2.1 Consulta de detalles sobre un gateway virtual.....	87
5.2.2 Actualización de un gateway virtual.....	93
5.2.3 Eliminación de un gateway virtual.....	101
5.2.4 Consulta de los gateway virtuales.....	104
5.2.5 Creación de un gateway virtual.....	112
5.3 Interfaz virtual.....	122
5.3.1 Consulta de detalles acerca de una interfaz virtual.....	122
5.3.2 Actualización de una interfaz virtual.....	133
5.3.3 Eliminación de una interfaz virtual.....	146
5.3.4 Consulta de la lista de interfaces virtuales.....	150

5.3.5 Creación de una interfaz virtual.....	163
5.3.6 Actualización de un par de interfaz virtual.....	178
5.3.7 Eliminación de un par de interfaz virtual.....	185
5.3.8 Creación de un par de interfaz virtual.....	189
5.3.9 Realización de una prueba de conmutación de interfaz virtual.....	196
5.3.10 Consulta de los registros de prueba de conmutación de una interfaz virtual.....	203
5.4 Gestión de etiquetas.....	209
5.4.1 Consulta de etiquetas por tipo de recurso.....	210
5.4.2 Consulta de etiquetas de recurso.....	214
5.4.3 Adición de una etiqueta de recurso.....	218
5.4.4 Adición o eliminación de etiquetas por lotes.....	223
5.4.5 Eliminación de una etiqueta de recurso.....	231
5.4.6 Consulta de recursos por etiqueta.....	234
5.5 Gestión de cuotas.....	251
5.5.1 Consulta de cuotas de recursos.....	251
6 Permisos y acciones admitidas.....	257
6.1 Introducción.....	257
6.2 Acciones apoyadas por la autorización basada en roles/políticas.....	258
7 Parámetros públicos.....	265
7.1 Códigos de estado comunes.....	265
7.2 Códigos de errores.....	266
A Apéndices.....	272
A.1 Obtención de un ID de proyecto.....	272

1 Antes de comenzar

1.1 Introducción

Bienvenido a la referencia de las API de Direct Connect. Direct Connect le permite conectar su centro de datos local a la nube a través de una conexión de red dedicada que ofrece un rendimiento estable, alta velocidad de transmisión, baja latencia de red y transmisión de datos segura. Direct Connect le permite maximizar las instalaciones de TI heredadas y aprovechar los servicios en la nube para crear un entorno de cómputo en la nube híbrida flexible y escalable.

Este documento describe cómo utilizar las interfaces de programación de aplicaciones (API) para realizar operaciones en los recursos de Direct Connect, como crear, consultar, modificar o eliminar conexiones. Para obtener más información acerca de todas las operaciones admitidas, consulte [Descripción de las API](#).

Antes de acceder a Direct Connect invocando a las API, familiarícese con los conceptos de productos. Para obtener más información, consulte [Descripción del servicio](#).

1.2 Invocación a API

Direct Connect admite las API de transferencia de estado representacional (REST) que se pueden invocar mediante HTTPS. Para obtener más información, véase [Llamada a APIs](#).

1.3 Puntos de conexión

Un punto de conexión es la **dirección de solicitud** para invocar a una API. Los puntos de conexión varían según los servicios y las regiones. For Direct Connect endpoints, consulte [Regiones y puntos de conexión](#).

1.4 Restricciones

El número de los recursos de Direct Connect que puede crear viene determinado por sus cuotas. Para ver o aumentar las cuotas, consulte la sección [Notas y limitaciones](#) en la *Guía de usuario de Direct Connect*.

Para obtener más restricciones, consulte las descripciones de las API.

2 Descripción de las API

Direct Connect proporciona las API de REST desarrolladas por Huawei.

Estas API le permiten usar todas las funciones de Direct Connect, por ejemplo, realizar operaciones en las conexiones, los gateway virtuales, las interfaces virtuales y las etiquetas.

Tabla 2-1 describe las API proporcionadas por Direct Connect.

Tabla 2-1 API de Direct Connect

Recurso	Las API
Conexiones	Las API para consultar detalles de conexión, actualizar una conexión, eliminar una conexión, consultar la lista de conexiones, consultar la lista de conexiones alojadas, crear una conexión alojada, consultar detalles de conexión alojada, actualizar información de conexión alojada y eliminar una conexión alojada
Gateway virtuales	Las API para consultar detalles del gateway virtual, actualizar un gateway virtual, eliminar un gateway virtual, consultar la lista de gateway virtual y crear un gateway virtual
Interfaces virtuales	Las API para consultar detalles de la interfaz virtual, actualizar una interfaz virtual, eliminar una interfaz virtual, consultar la lista de interfaz virtual y crear una interfaz virtual
Etiquetas	Las API para consultar etiquetas de proyecto, consultar etiquetas de recursos, agregar una etiqueta de recursos, agregar o eliminar etiquetas de recursos por lotes, eliminar una etiqueta de recursos y consultar recursos por etiqueta
Cuotas	API para consultar cuotas de recursos

3 Llamada a APIs

3.1 Hacer una solicitud de API

Esta sección describe la estructura de una solicitud de API de REST y utiliza la API de IAM para **obtener un token de usuario** como ejemplo para demostrar cómo llamar a una API. El token obtenido se puede usar entonces para autenticar la llamada de otras API.

Solicitud de URI

Un URI de solicitud tiene el siguiente formato:

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

Aunque se incluye un URI de solicitud en la cabecera de solicitud, la mayoría de los lenguajes de programación o marcos requieren que el URI de solicitud se transmita por separado.

Tabla 3-1 Descripción del parámetro de URI

Parámetro	Descripción
URI-scheme	Protocolo utilizado para transmitir solicitudes. Todas las API usan HTTPS.
Endpoint	Nombre de dominio o dirección IP del servidor que lleva el servicio REST. El punto de conexión varía entre los servicios en diferentes regiones. Se puede obtener del Regions y puntos de conexión . Por ejemplo, el punto final de IAM en la región CN-Hong Kong es iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Ruta de acceso de una API para realizar una operación especificada. Obtener la ruta de acceso desde el URI de una API. Por ejemplo, la resource-path de la API utilizada para obtener un token de usuario es /v3/auth/tokens .

Parámetro	Descripción
query-string	Parámetro de consulta, que es opcional. Asegúrese de que se incluye un signo de interrogación (?) antes de cada parámetro de consulta que tiene el formato de <i>Parameter name=Parameter value</i> . Por ejemplo, ?limit=10 indica que se mostrará un máximo de 10 registros de datos.

Por ejemplo, para obtener un token IAM en la región **CN-Hong Kong**, obtenga el punto final de IAM (iam.ap-southeast-1.myhuaweicloud.com) para esta región y la ruta de recursos (/v3/auth/tokens) en el URI de la API utilizada para **obtener un token de usuario**. A continuación, construya el URI de la siguiente manera:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figura 3-1 Ejemplo de URI



NOTA

Para simplificar la visualización de URI en este documento, cada API se proporciona solo con una **resource-path** y un método de solicitud. El **URI-scheme** de todas las API es **HTTPS**, y los puntos de conexión de todas las API en la misma región son idénticos.

Métodos de solicitud

El protocolo HTTP define los siguientes métodos de solicitud que se pueden usar para enviar una solicitud al servidor.

Tabla 3-2 Métodos de HTTP

Método	Descripción
GET	Solicita al servidor que devuelva los recursos especificados.
PUT	Solicita al servidor que actualice los recursos especificados.
POST	Solicita al servidor que agregue recursos o realice operaciones especiales.
DELETE	Solicita al servidor que elimine los recursos especificados, por ejemplo, un objeto.
HEAD	Igual que GET, excepto que el servidor debe devolver solo el encabezado de respuesta.

Método	Descripción
PATCH	Solicita al servidor que actualice el contenido parcial de un recurso especificado. Si el recurso no existe, se creará un nuevo recurso.

Por ejemplo, en el caso de la API usada para **obtener un token de usuario**, el método de solicitud es **POST**. La solicitud es la siguiente:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Encabezado de la solicitud

También puede agregar campos de encabezado adicionales a una solicitud, como los campos requeridos por un método URI o HTTP especificado. Por ejemplo, para solicitar la información de autenticación, agregue **Content-Type**, que especifica el tipo de cuerpo de la solicitud.

Los campos de encabezado de solicitud comunes son los siguientes.

Tabla 3-3 Campos de encabezado de solicitud comunes

Parámetro	Descripción	Obligatorio	Valor de ejemplo
Host	Especifica el nombre de dominio del servidor y el número de puerto de los recursos que se solicitan. El valor se puede obtener de la URL de la API de servicio. El valor tiene el formato de nombre de <i>Hostname:Port number</i> . Si no se especifica el número de puerto, se utiliza el puerto predeterminado. El número de puerto predeterminado para https es 443 .	No This field is mandatory for AK/SK authentication.	code.test.com o code.test.com:443
Content-Type	Especifica el tipo (o formato) del cuerpo del mensaje. Se recomienda el valor predeterminado application/json . Otros valores de este campo se proporcionarán para APIs específicas si los hay.	Sí	application/json
Content-Length	Especifica la longitud del cuerpo de la solicitud. La unidad es byte.	No	3495

Parámetro	Descripción	Obligatorio	Valor de ejemplo
X-Project-Id	Especifica el ID del proyecto. Obtenga el ID del proyecto siguiendo las instrucciones en Obtención de un ID de proyecto .	No Este campo es obligatorio para las solicitudes que usan autenticación AK/SK en el escenario de nube dedicada (DeC) o multiproyecto.	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Especifica el token de usuario. Es una respuesta a la API para obtener un token de usuario (Esta es la única API que no requiere autenticación). Después de procesar la solicitud, el valor de X-Subject-Token en el encabezado de respuesta es el valor de token.	No Este campo es obligatorio para la autenticación de tokens.	Lo siguiente es parte de un token de ejemplo: MIIPAgYJKoZIhvcNAQcCo...ggg1BB IINPXsidG9rZ

NOTA

Además de admitir la autenticación mediante tokens, las API admiten la autenticación mediante AK/SK, que utiliza SDK para firmar una solicitud. Durante la firma, los encabezados **Authorization** (autenticación de firma) y **X-Sdk-Date** (hora en que se envía una solicitud) se agregan automáticamente en la solicitud.

Para obtener más detalles, consulte "Autenticación usando AK/SK" en [Autenticación](#).

La API utilizada para [obtener un token de usuario](#) no requiere autenticación. Por lo tanto, solo es necesario agregar el campo **Content-Type** a las solicitudes para llamar a la API. Un ejemplo de tales solicitudes es el siguiente:

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

(Opcional) Cuerpo de solicitud

Esta parte es opcional. El cuerpo de una solicitud se envía a menudo en un formato estructurado como se especifica en el campo de encabezado **Content-Type**. El cuerpo de la solicitud transfiere contenido excepto el encabezado de la solicitud.

El cuerpo de la solicitud varía entre las API. Algunas API no requieren el cuerpo de la solicitud, como las API solicitadas mediante los métodos GET y DELETE.

En el caso de la API usada para [obtener un token de usuario](#), los parámetros de solicitud y la descripción del parámetro se pueden obtener a partir de la solicitud de API. A continuación se proporciona una solicitud de ejemplo con un cuerpo incluido. Reemplace *username*,

domainname, ******* (login password), and *xxxxxxxxxxxxxxxxxxxx* con los valores reales. Obtener un nombre de proyecto de [Regions y puntos de conexión](#).

📖 NOTA

El parámetro de **scope** especifica dónde surte efecto un token. Puede establecer **scope** para una cuenta o un proyecto en una cuenta. En el siguiente ejemplo, el token solo tiene efecto para los recursos de un proyecto especificado. Para obtener más información sobre esta API, consulte [Obtención de un token de usuario](#).

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

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

Si todos los datos necesarios para la solicitud de API están disponibles, puedes enviar la solicitud para llamar a la API a través de [curl](#), [Postman](#) o codificación. En la respuesta a la API utilizada para obtener un token de usuario, **x-subject-token** es el token de usuario deseado. Este token se puede utilizar para autenticar la llamada de otras API.

3.2 Autenticación

Las solicitudes para llamar a una API se pueden autenticar mediante cualquiera de los siguientes métodos:

- Autenticación de tokens: las solicitudes se autentican mediante tokens.
- Autenticación AK/SK: Las solicitudes se cifran utilizando pares AK/SK. Se recomienda la autenticación AK/SK porque es más segura que la autenticación de token.

Autenticación de tokens

📖 NOTA

El período de validez de un token es de 24 horas. Cuando utilice un token para la autenticación, guarde en caché para evitar llamar con frecuencia a la API de IAM utilizada para obtener un token de usuario.

Un token especifica los permisos temporales en un sistema informático. Durante la autenticación de API mediante un token, el token se agrega a las solicitudes para obtener permisos para llamar a la API. Puede obtener un token llamando a la [Obtención del token de usuario](#).

Un servicio en la nube se puede implementar como un servicio a nivel de proyecto o como un servicio global.

- Para un servicio a nivel de proyecto, necesita obtener un token a nivel de proyecto. Cuando llames a la API, configura **auth.scope** en el cuerpo de la solicitud para **project**.
- Para un servicio global, necesita obtener un token global. Cuando llames a la API, establece **auth.scope** en el cuerpo de la solicitud como **domain**.

IMS es un servicio a nivel de proyecto. Cuando llames a la API, configura **auth.scope** en el cuerpo de la solicitud para **project**.

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

Después de obtener un token, el campo de encabezado **X-Auth-Token** debe agregarse a las solicitudes para especificar el token al llamar a otras API. Por ejemplo, si el token es **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** se puede añadir a una solicitud de la siguiente manera:

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

Autenticación por AK/SK

NOTA

La autenticación AK/SK admite solicitudes de API con un cuerpo de no más de 12 MB. Para las solicitudes de API con un cuerpo más grande, se recomienda la autenticación de tokens.

En la autenticación AK/SK, AK/SK se utiliza para firmar solicitudes y la firma se agrega a continuación a las solicitudes de autenticación.

- AK: ID de clave de acceso, que es un identificador único usado junto con una clave de acceso secreta para firmar solicitudes criptográficamente.
- SK: clave de acceso secreta usada junto con un AK para firmar solicitudes criptográficamente. Identifica un remitente de la solicitud y evita que la solicitud sea modificada.

En la autenticación AK/SK, puede usar un AK/SK para firmar solicitudes basadas en el algoritmo de firma o usando el SDK de firma. Para obtener más información sobre cómo firmar solicitudes y usar el SDK de firma, consulte [Guía de firma de solicitudes de API](#).

 **NOTA**

El SDK de firma solo se utiliza para firmar solicitudes y es diferente de los SDK proporcionados por los servicios.

4 Uso de API

Direct Connect cuenta con los puntos de conexión independientes. Cuando se realizan invocaciones a la API a Direct Connect, se utilizan las API de Direct Connect. Por lo tanto, es necesario prestar atención especial a los puntos de conexión utilizados del servicio concreto cuando se invocan las API de Direct Connect.

5 Las API

5.1 Conexión

5.1.1 Consulta de detalles acerca de una conexión

Función

Esta API se utiliza para consultar detalles sobre una conexión.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-1 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
direct_connect_id	Sí	String	Especifica el ID de conexión. Mínimo: 36 Máximo: 36
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-2 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-3 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-4 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
direct_connect	DirectConnect object	Especifica los parámetros para crear una conexión.

Tabla 5-5 DirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.

Parámetro	Tipo	Descripción
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
port_type	String	Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G , 10G , 40G o 100G . Valores de enumeración: <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 100000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
device_id	String	Especifica el ID del dispositivo conectado a la conexión. Mínimo: 0 Máximo: 36

Parámetro	Tipo	Descripción
type	String	<p>Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo).</p> <p>Predeterminado: standard</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted
hosting_id	String	<p>Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.</p>
charge_mode	String	<p>Especifica el modo de facturación, que puede ser prepayment, bandwidth o traffic.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● prepayment ● bandwidth ● traffic
provider	String	<p>Especifica el portador de línea de una conexión. Puede ser China Telecom, China Unicom, China Mobile y otros operadores que proporcionan las líneas.</p>
admin_state_up	Boolean	<p>Especifica el estado administrativo, que puede ser true o false.</p> <p>Predeterminado: true</p>
vlan	Integer	<p>Especifica la VLAN asignada a la conexión alojada.</p> <p>Mínimo: 0</p> <p>Máximo: 3999</p>

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de la conexión. Las opciones son las siguientes: ACTIVE: La conexión está en el estado normal. DOWN: el puerto de la conexión está en estado inactivo, lo que puede causar fallas en la línea. BUILD: Se están realizando operaciones relacionadas con la conexión. ERROR: La configuración de la conexión es incorrecta. Póngase en contacto con el servicio de atención al cliente para rectificar la avería. PENDING_DELETE: se está eliminando la conexión. DELETED: Se ha eliminado la conexión. APPLY: se envía una solicitud de conexión. DENY: Una encuesta de sitio se rechaza porque el cliente no cumple con los requisitos. PENDING_PAY: El pedido de la conexión debe ser pagado. PAID: El pedido de la conexión ha sido pagado. PENDING_SURVEY: Se requiere una encuesta del sitio para la conexión.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● PAID ● APPLY ● PENDING_SURVEY ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● DELETED ● DENY ● PENDING_PAY
apply_time	String	Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
create_time	String	Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
peer_port_type	String	Especifica el tipo de puerto de otro extremo.
peer_provider	String	Especifica el portador conectado a la conexión.

Parámetro	Tipo	Descripción
order_id	String	Especifica el ID de pedido de conexión, que se utiliza para admitir la facturación basada en la duración e identificar los pedidos de usuario.
product_id	String	Especifica el ID de producto correspondiente al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
spec_code	String	Especifica las especificaciones del producto correspondientes al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
period_type	Integer	Especifica si una conexión en un pedido especificado se factura por año o por mes.
period_num	Integer	Especifica la duración del servicio requerida de una conexión anual/mensual.
vgw_type	String	Especifica el tipo de gateway requerido por una conexión directa. Predeterminado: default Valores de enumeración: ● default
lag_id	String	Especifica el ID del LAG al que pertenece la conexión.
signed_agreement_status	String	Especifica el estado de firma del acuerdo de Direct Connect. Valores de enumeración: ● signed
signed_agreement_time	String	Especifica la hora en que se firma el acuerdo de línea.
enterprise_project_id	String	Especifica el identificador del proyecto de empresa al que pertenece la conexión. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
locales	LocalesBody object	Especifica la región de la conexión.
support_feature	Array of strings	Muestra las características admitidas por la conexión.

Parámetro	Tipo	Descripción
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
email	String	Especifica la información de correo electrónico del cliente.
onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo. Este parámetro se utiliza en escenarios de ventas de línea.
building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos. Este parámetro se utiliza en escenarios de ventas de línea.
last_onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
last_building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
modified_bandwidth	Integer	Especifica el nuevo ancho de banda después de cambiar el ancho de banda de línea.
change_mode	Integer	Especifica el estado de un cambio de renovación.
onestopdc_statuses	String	Especifica el estado de una conexión de servicio completo.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
auto_renew	Integer	Especifica si se renovará automáticamente una suscripción anual/mensual.
ratio_95peak	Integer	Especifica el porcentaje del ancho de banda mínimo para la facturación del percentil 95. Mínimo: 0 Máximo: 100

Tabla 5-6 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Tabla 5-7 LocalesBody

Parámetro	Tipo	Descripción
en_us	String	Especifica el nombre de la región en inglés. Mínimo: 0 Máximo: 255
zh_cn	String	Especifica el nombre de la región en chino. Mínimo: 0 Máximo: 255

Ejemplo de las solicitudes

Consulta de detalles acerca de una conexión

```
GET https://{dc_endpoint}/v3/6f9e9263116a4b68818cf1edce16bc4f/dcaas/direct-connects/6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a
```

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- Se consultan los detalles de la conexión.

```
{
  "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" : "6f9e9263116a4b68818cf1edce16bc4f",
"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
}
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.1.2 Actualización de una conexión

Función

Esta API se utiliza para actualizar una conexión, incluida la actualización de su nombre y descripción.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-8 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
direct_connect_id	Sí	String	Especifica el ID de conexión. Mínimo: 36 Máximo: 36
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-9 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-10 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
direct_connect	No	UpdateDirectConnect object	Especifica los parámetros necesarios para actualizar una conexión.

Tabla 5-11 UpdateDirectConnect

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
bandwidth	No	Integer	Especifica el tamaño de ancho de banda de la conexión alojada en Mbit/s. Mínimo: 2 Máximo: 100000
peer_location	No	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
status	No	String	Especifica el estado del recurso, que puede ser PENDING_PAY o APPLY . Valores de enumeración: ● PENDING_PAY ● APPLY
provider_status	No	String	Especifica el estado del portador, que puede ser ACTIVE o DOWN . Valores de enumeración: ● ACTIVE ● DOWN

Parámetros de respuesta

Código de estado: 200

Tabla 5-12 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
direct_connect	DirectConnect object	Especifica los parámetros para crear una conexión.

Tabla 5-13 DirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
port_type	String	Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G , 10G , 40G o 100G . Valores de enumeración: <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 100000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255

Parámetro	Tipo	Descripción
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
device_id	String	Especifica el ID del dispositivo conectado a la conexión. Mínimo: 0 Máximo: 36
type	String	Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo). Predeterminado: standard Valores de enumeración: <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
charge_mode	String	Especifica el modo de facturación, que puede ser prepayment , bandwidth o traffic . Valores de enumeración: <ul style="list-style-type: none"> ● prepayment ● bandwidth ● traffic
provider	String	Especifica el portador de línea de una conexión. Puede ser China Telecom, China Unicom, China Mobile y otros operadores que proporcionan las líneas.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de la conexión. Las opciones son las siguientes: ACTIVE: La conexión está en el estado normal. DOWN: el puerto de la conexión está en estado inactivo, lo que puede causar fallas en la línea. BUILD: Se están realizando operaciones relacionadas con la conexión. ERROR: La configuración de la conexión es incorrecta. Póngase en contacto con el servicio de atención al cliente para rectificar la avería. PENDING_DELETE: se está eliminando la conexión. DELETED: Se ha eliminado la conexión. APPLY: se envía una solicitud de conexión. DENY: Una encuesta de sitio se rechaza porque el cliente no cumple con los requisitos. PENDING_PAY: El pedido de la conexión debe ser pagado. PAID: El pedido de la conexión ha sido pagado. PENDING_SURVEY: Se requiere una encuesta del sitio para la conexión.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● PAID ● APPLY ● PENDING_SURVEY ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● DELETED ● DENY ● PENDING_PAY
apply_time	String	Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
create_time	String	Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
peer_port_type	String	Especifica el tipo de puerto de otro extremo.
peer_provider	String	Especifica el portador conectado a la conexión.

Parámetro	Tipo	Descripción
order_id	String	Especifica el ID de pedido de conexión, que se utiliza para admitir la facturación basada en la duración e identificar los pedidos de usuario.
product_id	String	Especifica el ID de producto correspondiente al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
spec_code	String	Especifica las especificaciones del producto correspondientes al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
period_type	Integer	Especifica si una conexión en un pedido especificado se factura por año o por mes.
period_num	Integer	Especifica la duración del servicio requerida de una conexión anual/mensual.
vgw_type	String	Especifica el tipo de gateway requerido por una conexión directa. Predeterminado: default Valores de enumeración: ● default
lag_id	String	Especifica el ID del LAG al que pertenece la conexión.
signed_agreement_status	String	Especifica el estado de firma del acuerdo de Direct Connect. Valores de enumeración: ● signed
signed_agreement_time	String	Especifica la hora en que se firma el acuerdo de línea.
enterprise_project_id	String	Especifica el identificador del proyecto de empresa al que pertenece la conexión. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
locales	LocalesBody object	Especifica la región de la conexión.
support_feature	Array of strings	Muestra las características admitidas por la conexión.

Parámetro	Tipo	Descripción
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
email	String	Especifica la información de correo electrónico del cliente.
onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo. Este parámetro se utiliza en escenarios de ventas de línea.
building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos. Este parámetro se utiliza en escenarios de ventas de línea.
last_onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
last_building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
modified_bandwidth	Integer	Especifica el nuevo ancho de banda después de cambiar el ancho de banda de línea.
change_mode	Integer	Especifica el estado de un cambio de renovación.
onestopdc_statuses	String	Especifica el estado de una conexión de servicio completo.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
auto_renew	Integer	Especifica si se renovará automáticamente una suscripción anual/mensual.
ratio_95peak	Integer	Especifica el porcentaje del ancho de banda mínimo para la facturación del percentil 95. Mínimo: 0 Máximo: 100

Tabla 5-14 Etiqueta

Parámetro	Tipo	Descripción
key	String	<p>Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_).</p> <p>Mínimo: 0 Máximo: 36</p>
value	String	<p>Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.).</p> <p>Mínimo: 0 Máximo: 43</p>

Tabla 5-15 LocalesBody

Parámetro	Tipo	Descripción
en_us	String	<p>Especifica el nombre de la región en inglés.</p> <p>Mínimo: 0 Máximo: 255</p>
zh_cn	String	<p>Especifica el nombre de la región en chino.</p> <p>Mínimo: 0 Máximo: 255</p>

Ejemplo de las solicitudes

Actualizar el nombre y la descripción de una conexión

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

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

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- La conexión se actualiza.

```
{
  "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" : "6f9263116a4b68818cf1edce16bc4f",
  "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
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Actualizar el nombre y la descripción de una conexión

```
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.setName("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

Actualizar el nombre y la descripción de una conexión

```
# 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

Actualizar el nombre y la descripción de una conexión

```
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)
}
}

```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.1.3 Eliminación de una conexión

Función

Esta API se utiliza solo para eliminar conexiones de pago por uso. Para eliminar conexiones anuales/mensuales, primero debe cancelar su suscripción.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-16 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
direct_connect_id	Sí	String	Especifica el ID de conexión. Mínimo: 36 Máximo: 36
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-17 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

No hay

Ejemplo de solicitudes

Eliminación de una conexión

```
DELETE https://{dc_endpoint}/v3/6f9e9263116a4b68818cf1edce16bc4f/dcaas/direct-connects/6ecd9cf3-ca64-46c7-863f-f2eb1b9e838a
```

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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 huaweicloudskdkc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskdkc.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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.1.4 Consulta de la lista de conexiones

Función

Esta API se utiliza para consultar todas las conexiones directas creadas por un tenant.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-18 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-19 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000
marker	No	String	Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit . Mínimo: 0 Máximo: 36
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5
sort_key	No	String	Especifica el campo de la ordenación. Predeterminado: id Mínimo: 0 Máximo: 36

Parámetro	Obligatorio	Tipo	Descripción
sort_dir	No	Array	Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc .
hosting_id	No	Array	Especifica el identificador de conexión de operaciones por el que se filtran las conexiones alojadas. Longitud del arreglo: 0 - 5
enterprise_project_id	No	Array	Filtra instancias de recursos por ID de proyecto de empresa. Longitud del arreglo: 1 - 10
id	No	Array	Especifica el ID de recurso por el que se filtran las instancias. Longitud del arreglo: 1 - 5
name	No	Array	Especifica el nombre del recurso por el que se filtran las instancias. Puede especificar varios nombres. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-20 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-21 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
direct_connects	Array of DirectConnect objects	Especifica los parámetros para crear una conexión.
page_info	PageInfo object	Especifica la información de consulta de paginación.

Tabla 5-22 DirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
port_type	String	Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G , 10G , 40G o 100G . Valores de enumeración: <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 100000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255

Parámetro	Tipo	Descripción
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
device_id	String	Especifica el ID del dispositivo conectado a la conexión. Mínimo: 0 Máximo: 36
type	String	Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo). Predeterminado: standard Valores de enumeración: <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
charge_mode	String	Especifica el modo de facturación, que puede ser prepayment , bandwidth o traffic . Valores de enumeración: <ul style="list-style-type: none"> ● prepayment ● bandwidth ● traffic
provider	String	Especifica el portador de línea de una conexión. Puede ser China Telecom, China Unicom, China Mobile y otros operadores que proporcionan las líneas.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de la conexión. Las opciones son las siguientes: ACTIVE: La conexión está en el estado normal. DOWN: el puerto de la conexión está en estado inactivo, lo que puede causar fallas en la línea. BUILD: Se están realizando operaciones relacionadas con la conexión. ERROR: La configuración de la conexión es incorrecta. Póngase en contacto con el servicio de atención al cliente para rectificar la avería. PENDING_DELETE: se está eliminando la conexión. DELETED: Se ha eliminado la conexión. APPLY: se envía una solicitud de conexión. DENY: Una encuesta de sitio se rechaza porque el cliente no cumple con los requisitos. PENDING_PAY: El pedido de la conexión debe ser pagado. PAID: El pedido de la conexión ha sido pagado. PENDING_SURVEY: Se requiere una encuesta del sitio para la conexión.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● PAID ● APPLY ● PENDING_SURVEY ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● DELETED ● DENY ● PENDING_PAY
apply_time	String	Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
create_time	String	Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
peer_port_type	String	Especifica el tipo de puerto de otro extremo.
peer_provider	String	Especifica el portador conectado a la conexión.

Parámetro	Tipo	Descripción
order_id	String	Especifica el ID de pedido de conexión, que se utiliza para admitir la facturación basada en la duración e identificar los pedidos de usuario.
product_id	String	Especifica el ID de producto correspondiente al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
spec_code	String	Especifica las especificaciones del producto correspondientes al pedido de la conexión, que se utiliza para las políticas de facturación personalizadas, como los paquetes basados en la duración.
period_type	Integer	Especifica si una conexión en un pedido especificado se factura por año o por mes.
period_num	Integer	Especifica la duración del servicio requerida de una conexión anual/mensual.
vgw_type	String	Especifica el tipo de gateway requerido por una conexión directa. Predeterminado: default Valores de enumeración: ● default
lag_id	String	Especifica el ID del LAG al que pertenece la conexión.
signed_agreement_status	String	Especifica el estado de firma del acuerdo de Direct Connect. Valores de enumeración: ● signed
signed_agreement_time	String	Especifica la hora en que se firma el acuerdo de línea.
enterprise_project_id	String	Especifica el identificador del proyecto de empresa al que pertenece la conexión. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
locales	LocalesBody object	Especifica la región de la conexión.
support_feature	Array of strings	Muestra las características admitidas por la conexión.

Parámetro	Tipo	Descripción
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
email	String	Especifica la información de correo electrónico del cliente.
onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo. Este parámetro se utiliza en escenarios de ventas de línea.
building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos. Este parámetro se utiliza en escenarios de ventas de línea.
last_onestop_product_id	String	Especifica el ID de producto de una conexión de servicio completo antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
last_building_line_product_id	String	Especifica el ID de producto del recurso de línea utilizado en la sala de equipos antes del cambio. Este parámetro se utiliza en escenarios de venta de línea y se utiliza para guardar el último registro cuando se cambia el ancho de banda de línea.
modified_bandwidth	Integer	Especifica el nuevo ancho de banda después de cambiar el ancho de banda de línea.
change_mode	Integer	Especifica el estado de un cambio de renovación.
onestopdc_statuses	String	Especifica el estado de una conexión de servicio completo.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
auto_renew	Integer	Especifica si se renovará automáticamente una suscripción anual/mensual.
ratio_95peak	Integer	Especifica el porcentaje del ancho de banda mínimo para la facturación del percentil 95. Mínimo: 0 Máximo: 100

Tabla 5-23 Etiqueta

Parámetro	Tipo	Descripción
key	String	<p>Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_).</p> <p>Mínimo: 0 Máximo: 36</p>
value	String	<p>Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.).</p> <p>Mínimo: 0 Máximo: 43</p>

Tabla 5-24 LocalesBody

Parámetro	Tipo	Descripción
en_us	String	<p>Especifica el nombre de la región en inglés.</p> <p>Mínimo: 0 Máximo: 255</p>
zh_cn	String	<p>Especifica el nombre de la región en chino.</p> <p>Mínimo: 0 Máximo: 255</p>

Tabla 5-25 PageInfo

Parámetro	Tipo	Descripción
previous_marker	String	<p>Especifica el marcador de la página anterior. El valor es el UUID del recurso.</p> <p>Mínimo: 0 Máximo: 36</p>
current_count	Integer	<p>Especifica el número de recursos de la lista actual.</p> <p>Mínimo: 0 Máximo: 2000</p>

Parámetro	Tipo	Descripción
next_marker	String	<p>Especifica el marcador de la página siguiente. El valor es el UUID del recurso. Si el valor está vacío, el recurso se encuentra en la última página.</p> <p>Mínimo: 0 Máximo: 36</p>

Ejemplo de las solicitudes

Consulta de la lista de conexiones

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

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- Se solicita la lista de conexiones.

```
{
  "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  
} ]  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	OK

Códigos de error

Consulte [Códigos de error](#).

5.1.5 Consulta de la lista de conexiones alojadas

Función

Esta API se utiliza para consultar las conexiones alojadas creadas por socios.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-26 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-27 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000
marker	No	String	Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit . Mínimo: 0 Máximo: 36
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5
sort_dir	No	Array	Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc .
sort_key	No	String	Especifica el campo de la ordenación. Predeterminado: id Mínimo: 0 Máximo: 36

Parámetro	Obligatorio	Tipo	Descripción
hosting_id	No	Array	Especifica el identificador de conexión de operaciones por el que se filtran las conexiones alojadas. Longitud del arreglo: 0 - 5
id	No	Array	Especifica el ID de recurso por el que se filtran las instancias. Longitud del arreglo: 1 - 5
name	No	Array	Especifica el nombre del recurso por el que se filtran las instancias. Puede especificar varios nombres. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-28 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-29 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de solicitud de la operación actual.
hosted_connects	Array of HostedDirectConnect objects	Especifica los parámetros para crear una conexión alojada.

Parámetro	Tipo	Descripción
page_info	PageInfo object	Especifica la información de consulta de paginación.

Tabla 5-30 HostedDirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 400000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
provider	String	Especifica el proveedor de la línea arrendada.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true

Parámetro	Tipo	Descripción
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999
status	String	Especifica el estado de funcionamiento de la conexión alojada. Las opciones son las siguientes: <ul style="list-style-type: none"> ● BUILD: Se ha creado la conexión alojada. ● ACTIVE: El gateway virtual asociado es normal. ● DOWN: El puerto utilizado por la conexión alojada está inactivo, lo que indica que puede haber fallos de línea. ● ERROR: El gateway virtual asociado es anormal. ● PENDING_DELETE: se está eliminando la conexión alojada. ● PENDING_UPDATE se está actualizando la conexión alojada. ● PENDING_CREATE: Se está creando la conexión alojada. Valores de enumeración: <ul style="list-style-type: none"> ● BUILD ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● PENDING_UPDATE ● PENDING_CREATE
apply_time	String	Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
create_time	String	Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ .
provider_status	String	Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN . Valores de enumeración: <ul style="list-style-type: none"> ● ACTIVE ● DOWN

Parámetro	Tipo	Descripción
port_type	String	<p>Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G, 10G, 40G o 100G.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G
type	String	<p>Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo).</p> <p>Predeterminado: standard</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted

Tabla 5-31 PageInfo

Parámetro	Tipo	Descripción
previous_marker	String	<p>Especifica el marcador de la página anterior. El valor es el UUID del recurso.</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>
current_count	Integer	<p>Especifica el número de recursos de la lista actual.</p> <p>Mínimo: 0</p> <p>Máximo: 2000</p>
next_marker	String	<p>Especifica el marcador de la página siguiente. El valor es el UUID del recurso. Si el valor está vacío, el recurso se encuentra en la última página.</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>

Ejemplo de solicitudes

Consultar la lista de conexiones alojadas creadas por un socio

```
POST https://{dc_endpoint}/v3/6f8e9263116a4b68818cf1edce16bc4f/dcaas/hosted-connects
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

- Se consulta la lista de conexiones alojadas.

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

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.1.6 Creación de una conexión alojada

Función

Esta API es utilizada por los socios para crear conexiones alojadas para sus usuarios. El creador debe tener la calificación de socio y tener una conexión de operaciones.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-32 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-33 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-34 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
hosted_connect	Sí	CreateHostedDirectConnect object	Especifica los parámetros para crear una conexión alojada.

Tabla 5-35 CreateHostedDirectConnect

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre de la conexión alojada. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información adicional acerca de la conexión alojada. Mínimo: 0 Máximo: 128
bandwidth	Sí	Integer	Especifica el tamaño de ancho de banda de la conexión alojada en Mbit/s. Mínimo: 2 Máximo: 400000
hosting_id	Sí	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
vlan	Sí	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999
resource_tenant_id	Sí	String	Especifica el ID de proyecto del tenant especificado para el que se va a crear una conexión alojada. Mínimo: 0 Máximo: 32

Parámetro	Obligatorio	Tipo	Descripción
peer_location	No	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, específica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255

Parámetros de respuesta

Código de estado: **201**

Tabla 5-36 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
hosted_connect	HostedDirectConnect object	Especifica los parámetros para crear una conexión alojada.

Tabla 5-37 HostedDirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 400000

Parámetro	Tipo	Descripción
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
provider	String	Especifica el proveedor de la línea arrendada.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de funcionamiento de la conexión alojada. Las opciones son las siguientes:</p> <ul style="list-style-type: none"> ● BUILD: Se ha creado la conexión alojada. ● ACTIVE: El gateway virtual asociado es normal. ● DOWN: El puerto utilizado por la conexión alojada está inactivo, lo que indica que puede haber fallos de línea. ● ERROR: El gateway virtual asociado es anormal. ● PENDING_DELETE: se está eliminando la conexión alojada. ● PENDING_UPDATE se está actualizando la conexión alojada. ● PENDING_CREATE: Se está creando la conexión alojada. <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● PENDING_UPDATE ● PENDING_CREATE
apply_time	String	<p>Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
create_time	String	<p>Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
port_type	String	<p>Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G, 10G, 40G o 100G.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G

Parámetro	Tipo	Descripción
type	String	<p>Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo).</p> <p>Predeterminado: standard</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted

Ejemplo de solicitudes

Creación de una conexión de alojamiento con el ancho de banda establecido a 10 Mbit/s y VLAN a 441

```
POST https://{dc_endpoint}/v3/6fbe9263116a4b68818cf1edc16bc4f/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
  }
}
```

Ejemplo de respuestas

Código de estado: 201

Creado

- Se crea la conexión alojada.

```
{
  "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" : "a59a3776faald055f8124dc7b0977a90"  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Creación de una conexión de alojamiento con el ancho de banda establecido a 10 Mbit/s y VLAN a 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

Creación de una conexión de alojamiento con el ancho de banda establecido a 10 Mbit/s y VLAN a 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

Creación de una conexión de alojamiento con el ancho de banda establecido a 10 Mbit/s y VLAN a 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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
201	Creado

Códigos de error

Consulte [Códigos de error](#).

5.1.7 Consulta de detalles sobre una conexión alojada

Función

Esta API se utiliza para consultar una conexión alojada de un socio.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-38 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
hosted_connect_id	Sí	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36

Tabla 5-39 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000

Parámetro	Obligatorio	Tipo	Descripción
marker	No	String	<p>Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit.</p> <p>Mínimo: 0 Máximo: 36</p>
fields	No	Array	<p>Especifica la lista de campos que se mostrarán.</p> <p>Longitud del arreglo: 1 - 5</p>
sort_dir	No	Array	<p>Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc.</p>
sort_key	No	String	<p>Especifica el campo de la ordenación.</p> <p>Predeterminado: id</p> <p>Mínimo: 0 Máximo: 36</p>
hosting_id	No	Array	<p>Especifica el identificador de conexión de operaciones por el que se filtran las conexiones alojadas.</p> <p>Longitud del arreglo: 0 - 5</p>

Parámetros de solicitud

Tabla 5-40 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-41 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
hosted_connect	HostedDirectConnect object	Especifica los parámetros para crear una conexión alojada.

Tabla 5-42 HostedDirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64

Parámetro	Tipo	Descripción
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 400000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
provider	String	Especifica el proveedor de la línea arrendada.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de funcionamiento de la conexión alojada. Las opciones son las siguientes:</p> <ul style="list-style-type: none"> ● BUILD: Se ha creado la conexión alojada. ● ACTIVE: El gateway virtual asociado es normal. ● DOWN: El puerto utilizado por la conexión alojada está inactivo, lo que indica que puede haber fallos de línea. ● ERROR: El gateway virtual asociado es anormal. ● PENDING_DELETE: se está eliminando la conexión alojada. ● PENDING_UPDATE se está actualizando la conexión alojada. ● PENDING_CREATE: Se está creando la conexión alojada. <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● PENDING_UPDATE ● PENDING_CREATE
apply_time	String	<p>Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
create_time	String	<p>Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
port_type	String	<p>Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G, 10G, 40G o 100G.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G

Parámetro	Tipo	Descripción
type	String	<p>Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo).</p> <p>Predeterminado: standard</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted

Ejemplo de solicitudes

Consultar detalles sobre una conexión alojada creada por un socio

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

Ejemplo de respuestas

Código de estado: 200

Aceptar

- Se consultan los detalles de la conexión alojada.

```
{
  "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" : "a59a3776faald055f8124dc7b0977a90"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)  
    }  
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.1.8 Actualización de una conexión alojada

Función

Actualización de una conexión alojada por un socio

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-43 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetro	Obligatorio	Tipo	Descripción
hosted_connect_id	Sí	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36

Parámetros de solicitud

Tabla 5-44 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-45 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
hosted_connect	No	UpdateHostedDirectConnect object	Especifica los parámetros necesarios para actualizar una conexión alojada.

Tabla 5-46 UpdateHostedDirectConnect

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128

Parámetro	Obligatorio	Tipo	Descripción
bandwidth	No	Integer	Especifica el tamaño de ancho de banda de la conexión alojada en Mbit/s. Mínimo: 2 Máximo: 400000
peer_location	No	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255

Parámetros de respuesta

Código de estado: 200

Tabla 5-47 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
hosted_connect	HostedDirectConnect object	Especifica los parámetros para crear una conexión alojada.

Tabla 5-48 HostedDirectConnect

Parámetro	Tipo	Descripción
id	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia.
name	String	Especifica el nombre de la conexión. Mínimo: 0 Máximo: 64

Parámetro	Tipo	Descripción
description	String	Proporciona la información complementaria sobre la conexión. Mínimo: 0 Máximo: 128
bandwidth	Integer	Especifica el ancho de banda de conexión en Mbit/s. Mínimo: 2 Máximo: 400000
location	String	Especifica la información acerca de la ubicación de Direct Connect. Mínimo: 0 Máximo: 255
peer_location	String	Especifica la ubicación de la instalación local en el otro extremo de la conexión, especifica del nombre de la calle o del centro de datos. Mínimo: 0 Máximo: 255
hosting_id	String	Especifica el identificador de la conexión de operaciones en la que se crea la conexión alojada.
provider	String	Especifica el proveedor de la línea arrendada.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
vlan	Integer	Especifica la VLAN asignada a la conexión alojada. Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
status	String	<p>Especifica el estado de funcionamiento de la conexión alojada. Las opciones son las siguientes:</p> <ul style="list-style-type: none"> ● BUILD: Se ha creado la conexión alojada. ● ACTIVE: El gateway virtual asociado es normal. ● DOWN: El puerto utilizado por la conexión alojada está inactivo, lo que indica que puede haber fallos de línea. ● ERROR: El gateway virtual asociado es anormal. ● PENDING_DELETE: se está eliminando la conexión alojada. ● PENDING_UPDATE se está actualizando la conexión alojada. ● PENDING_CREATE: Se está creando la conexión alojada. <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● BUILD ● ACTIVE ● DOWN ● ERROR ● PENDING_DELETE ● PENDING_UPDATE ● PENDING_CREATE
apply_time	String	<p>Especifica cuándo se solicitó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
create_time	String	<p>Especifica cuándo se creó la conexión. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ.</p>
provider_status	String	<p>Especifica el estado de la línea arrendada del portador. El estado puede ser ACTIVE o DOWN.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● ACTIVE ● DOWN
port_type	String	<p>Especifica el tipo de puerto utilizado por la conexión. El valor puede ser 1G, 10G, 40G o 100G.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● 1G ● 10G ● 40G ● 100G

Parámetro	Tipo	Descripción
type	String	<p>Especifica el tipo de conexión. El valor puede ser standard (una conexión estándar), hosting (una conexión de operaciones) o hosted (una conexión de servicio completo).</p> <p>Predeterminado: standard</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● standard ● hosting ● hosted ● onestop_standard ● onestop_hosted

Ejemplo de solicitudes

Actualizar el nombre y la descripción de una conexión alojada

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

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

Ejemplo de respuestas

Código de estado: 200

OK

- La conexión alojada se actualiza.

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Actualizar el nombre y la descripción de una conexión alojada

```
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

Actualizar el nombre y la descripción de una conexión alojada

```
# 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

Actualizar el nombre y la descripción de una conexión alojada

```
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)
}
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	OK

Códigos de error

Consulte [Códigos de error](#).

5.1.9 Eliminación de una conexión alojada

Función

Los socios utilizan esta API para eliminar las conexiones alojadas.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-49 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
hosted_connect_id	Sí	String	Especifica el ID de conexión alojada. Mínimo: 36 Máximo: 36

Parámetros de solicitud

Tabla 5-50 Parámetros de cabecera de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

No hay

Ejemplo de las solicitudes

Eliminación de una conexión alojada

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

Ejemplo de las respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.2 Gateway virtual

5.2.1 Consulta de detalles sobre un gateway virtual

Función

Esta API se utiliza para consultar detalles sobre un gateway virtual especificado.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-51 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
virtual_gateway_id	Sí	String	Especifica el ID del gateway virtual.

Tabla 5-52 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-53 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-54 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
virtual_gateway	VirtualGateway y object	Especifica los parámetros para crear un gateway virtual.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-55 VirtualGateway

Parámetro	Tipo	Descripción
id	String	Especifica el ID del gateway virtual.
vpc_id	String	Especifica el ID de la VPC conectada por el gateway virtual.

Parámetro	Tipo	Descripción
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia. Mínimo: 32 Máximo: 32
name	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
type	String	Especifica el tipo del gateway virtual. El valor solo puede ser default . Predeterminado: default
local_ep_group	Array of strings	Especifica las subredes IPv4 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC.
local_ep_group_ipv6	Array of strings	Especifica las subredes IPv6 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC. Este es un campo reservado.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
status	String	Especifica el estado del gateway virtual, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE o PENDING_DELETE .
bgp_asn	Integer	Especifica el ASN de BGP local del gateway virtual. Mínimo: 1 Máximo: 4294967295
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece el gateway virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
redundant_device_id	String	Especifica el ID del dispositivo redundante.

Parámetro	Tipo	Descripción
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-56 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Ejemplo de las solicitudes

Consulta de un gateway virtual

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

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- Se consultan los detalles del gateway virtual asociado con una VPC.

```
{
  "virtual_gateway" : {
    "id" : "20082c1b-3c99-48d8-8e8c-116af5d7e9f0",
    "name" : "vgw-c7b22",
    "description" : "",
    "tenant_id" : "0605768a3300d5762f82c01180692873",
    "vpc_id" : "6592c28e-95d7-4b0a-9f61-004fd03420c",
    "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
}
}

```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.2.2 Actualización de un gateway virtual

Función

Esta API se utiliza para actualizar un gateway virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-57 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
virtual_gateway_id	Sí	String	Especifica el ID del gateway virtual.

Parámetros de solicitud

Tabla 5-58 Parámetros de cabecera de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-59 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
virtual_gateway	No	UpdateVirtualGateway object	Especifica el gateway virtual que se va a actualizar.

Tabla 5-60 UpdateVirtualGateway

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
local_ep_group	No	Array of strings	Muestra las subredes IPv4 a las que se puede acceder a través del gateway virtual. Generalmente, la lista contiene los bloques CIDR de subred de la VPC asociada. Longitud del arreglo: 1 - 200

Parámetro	Obligatorio	Tipo	Descripción
local_ep_group_ipv6	No	Array of strings	Muestra las subredes IPv6 a las que se puede acceder mediante el gateway virtual. Por lo general, las subredes son las de la VPC asociada con el gateway virtual. Longitud del arreglo: 1 - 50

Parámetros de respuesta

Código de estado: **200**

Tabla 5-61 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
virtual_gateway	VirtualGateway y object	Especifica los parámetros para crear un gateway virtual.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-62 VirtualGateway

Parámetro	Tipo	Descripción
id	String	Especifica el ID del gateway virtual.
vpc_id	String	Especifica el ID de la VPC conectada por el gateway virtual.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia. Mínimo: 32 Máximo: 32
name	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
type	String	Especifica el tipo del gateway virtual. El valor solo puede ser default . Predeterminado: default

Parámetro	Tipo	Descripción
local_ep_group	Array of strings	Especifica las subredes IPv4 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC.
local_ep_group_ipv6	Array of strings	Especifica las subredes IPv6 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC. Este es un campo reservado.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
status	String	Especifica el estado del gateway virtual, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE o PENDING_DELETE .
bgp_asn	Integer	Especifica el ASN de BGP local del gateway virtual. Mínimo: 1 Máximo: 4294967295
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece el gateway virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
redundant_device_id	String	Especifica el ID del dispositivo redundante.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-63 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36

Parámetro	Tipo	Descripción
value	String	<p>Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.).</p> <p>Mínimo: 0</p> <p>Máximo: 43</p>

Ejemplo de las solicitudes

Actualizar el nombre y la descripción de un gateway virtual y cambiar su subred IPv4 a 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" ]
  }
}
```

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- Se actualiza el gateway virtual asociado con una VPC.

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Actualizar el nombre y la descripción de un gateway virtual y cambiar su subred IPv4 a 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.setName("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

Actualizar el nombre y la descripción de un gateway virtual y cambiar su subred IPv4 a 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

Actualizar el nombre y la descripción de un gateway virtual y cambiar su subred IPv4 a 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)
}
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.2.3 Eliminación de un gateway virtual

Función

La API se utiliza para eliminar un gateway virtual especificada.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-64 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
virtual_gateway_id	Sí	String	Especifica el ID del gateway virtual.

Parámetros de solicitud

Tabla 5-65 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

No hay

Ejemplo de solicitudes

Eliminación de un gateway virtual

```
DELETE https://{dc_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/dcaas/virtual-gateways/20082c1b-3c99-48d8-8e8c-116af5d7e9f0
```

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)  
}  
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.2.4 Consulta de los gateway virtuales

Función

Esta API se utiliza para consultar los gateway virtuales.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-66 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-67 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000
marker	No	String	Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit . Mínimo: 0 Máximo: 36
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5
sort_dir	No	Array	Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc .
sort_key	No	String	Especifica el campo de la ordenación. Predeterminado: id Mínimo: 0 Máximo: 36
id	No	Array	Especifica el ID de recurso por el que se filtran las instancias. Longitud del arreglo: 1 - 5
enterprise_project_id	No	Array	Filtra instancias de recursos por ID de proyecto de empresa. Longitud del arreglo: 1 - 10
vpc_id	No	Array	Especifica el ID de VPC por el que se filtran los gateway virtuales.

Parámetros de solicitud

Tabla 5-68 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-69 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
virtual_gateways	Array of VirtualGateway objects	Especifica los parámetros para crear un gateway virtual.
page_info	PageInfo object	Especifica la información de consulta de paginación.

Tabla 5-70 VirtualGateway

Parámetro	Tipo	Descripción
id	String	Especifica el ID del gateway virtual.
vpc_id	String	Especifica el ID de la VPC conectada por el gateway virtual.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia. Mínimo: 32 Máximo: 32

Parámetro	Tipo	Descripción
name	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
type	String	Especifica el tipo del gateway virtual. El valor solo puede ser default . Predeterminado: default
local_ep_group	Array of strings	Especifica las subredes IPv4 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC.
local_ep_group_ipv6	Array of strings	Especifica las subredes IPv6 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC. Este es un campo reservado.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
status	String	Especifica el estado del gateway virtual, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE o PENDING_DELETE .
bgp_asn	Integer	Especifica el ASN de BGP local del gateway virtual. Mínimo: 1 Máximo: 4294967295
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece el gateway virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
redundant_device_id	String	Especifica el ID del dispositivo redundante.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-71 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Tabla 5-72 PageInfo

Parámetro	Tipo	Descripción
previous_marker	String	Especifica el marcador de la página anterior. El valor es el UUID del recurso. Mínimo: 0 Máximo: 36
current_count	Integer	Especifica el número de recursos de la lista actual. Mínimo: 0 Máximo: 2000
next_marker	String	Especifica el marcador de la página siguiente. El valor es el UUID del recurso. Si el valor está vacío, el recurso se encuentra en la última página. Mínimo: 0 Máximo: 36

Ejemplo de las solicitudes

Consulta de la lista del gateway virtual

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

Ejemplo de las respuestas

Código de estado: 200

Aceptar

- Se consulta la lista de gateway virtuales asociadas con las VPC.

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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 huaweicloudskdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskdc.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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.2.5 Creación de un gateway virtual

Función

Esta API se utiliza para crear un gateway virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-73 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-74 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-75 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
virtual_gateway	No	CreateVirtualGateway object	Especifica los parámetros para crear un gateway virtual.

Tabla 5-76 CreateVirtualGateway

Parámetro	Obligatorio	Tipo	Descripción
vpc_id	Sí	String	Especifica el ID de la VPC con la que está asociada el gateway virtual.
name	No	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
local_ep_group	Sí	Array of strings	Muestra las subredes IPv4 a las que se puede acceder a través del gateway virtual. Generalmente, la lista contiene los bloques CIDR de subred de la VPC asociada.
local_ep_group_ipv6	No	Array of strings	Especifica las subredes IPv6 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC. Este es un campo reservado.
bgp_asn	No	Integer	Especifica el número del sistema autónomo (ASN) BGP local del gateway virtual. Mínimo: 1 Máximo: 4294967295
enterprise_project_id	No	String	Especifica el ID del proyecto de empresa al que pertenece el gateway virtual. Mínimo: 36 Máximo: 36
tags	No	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-77 Etiqueta

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	No	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Parámetros de respuesta

Código de estado: 201

Tabla 5-78 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
virtual_gateway	VirtualGateway y object	Especifica los parámetros para crear un gateway virtual.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-79 VirtualGateway

Parámetro	Tipo	Descripción
id	String	Especifica el ID del gateway virtual.
vpc_id	String	Especifica el ID de la VPC conectada por el gateway virtual.
tenant_id	String	Especifica el ID del proyecto al que pertenece la instancia. Mínimo: 32 Máximo: 32

Parámetro	Tipo	Descripción
name	String	Especifica el nombre del gateway virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona la información complementaria sobre el gateway virtual. Mínimo: 0 Máximo: 128
type	String	Especifica el tipo del gateway virtual. El valor solo puede ser default . Predeterminado: default
local_ep_group	Array of strings	Especifica las subredes IPv4 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC.
local_ep_group_ipv6	Array of strings	Especifica las subredes IPv6 conectadas por el gateway virtual, que suele ser los bloques CIDR de una VPC. Este es un campo reservado.
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false . Predeterminado: true
status	String	Especifica el estado del gateway virtual, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE o PENDING_DELETE .
bgp_asn	Integer	Especifica el ASN de BGP local del gateway virtual. Mínimo: 1 Máximo: 4294967295
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece el gateway virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
redundant_device_id	String	Especifica el ID del dispositivo redundante.
public_border_group	String	Especifica el grupo de bordes públicos de la AZ, que indica si el sitio es un sitio HomeZones.
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-80 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Ejemplo de las solicitudes

- Creación de un gateway virtual que se utilizará para acceder a una VPC, con el ASN de BGP local establecido en 64512 y la subred IPv4 en 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" ]
  }
}
```

Ejemplo de las respuestas

Código de estado: 201

Creado

- Se crea el gateway virtual para acceder a la VPC

```
{
  "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"  
}  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

- Creación de un gateway virtual que se utilizará para acceder a una VPC, con el ASN de BGP local establecido en 64512 y la subred IPv4 en 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());
    }
}
}
```

- **Creación de un gateway virtual que se conectará a un router empresarial y establecer el ASN de BGP en 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.getErrMsg());
    }
}
```

Python

- Creación de un gateway virtual que se utilizará para acceder a una VPC, con el ASN de BGP local establecido en 64512 y la subred IPv4 en 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)
```

- Creación de un gateway virtual que se conectará a un router empresarial y establecer el ASN de BGP en 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

- Creación de un gateway virtual que se utilizará para acceder a una VPC, con el ASN de BGP local establecido en 64512 y la subred IPv4 en 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 listLocalEpgGroupVirtualGateway = []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,
    LocalEpgGroup: listLocalEpgGroupVirtualGateway,
    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)
}
}
```

- Creación de un gateway virtual que se conectará a un router empresarial y establecer el ASN de BGP en 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)
}
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
201	Creado

Códigos de error

Consulte [Códigos de error](#).

5.3 Interfaz virtual

5.3.1 Consulta de detalles acerca de una interfaz virtual

Función

Esta API se utiliza para consultar detalles sobre una interfaz virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-81 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetro	Obligatorio	Tipo	Descripción
virtual_interfac e_id	Sí	String	Especifica el ID de la interfaz virtual. Mínimo: 36 Máximo: 36

Tabla 5-82 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-83 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-84 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
virtual_interfac e	VirtualInterfac e object	Especifica los parámetros para crear una interfaz virtual.

Tabla 5-85 VirtualInterface

Parámetro	Tipo	Descripción
id	String	Especifica el ID de la interfaz virtual. Máximo: 36
name	String	Especifica el nombre de la interfaz virtual. Máximo: 64
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false .
bandwidth	Integer	Especifica el ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647
create_time	String	Especifica la hora en que se creó la interfaz virtual. Se utiliza el formato de hora UTC aaa-MM-ddTHH:mm:ss.SSSZ . Máximo: 255
description	String	Proporciona la información complementaria sobre la interfaz virtual. Máximo: 128
direct_connect_id	String	Especifica el ID de conexión. Máximo: 36
service_type	String	Especifica el tipo de gateway, que puede ser VGW , GDGW o LGW . Valores de enumeración: <ul style="list-style-type: none"> ● VGW ● GDGW ● LGW
status	String	Especifica el estado de funcionamiento, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE , PENDING_DELETE , DELETED , AUTHORIZATION o REJECTED .
tenant_id	String	Especifica el ID del proyecto. Mínimo: 32 Máximo: 32

Parámetro	Tipo	Descripción
type	String	Especifica el tipo de la interfaz virtual. El valor es private . Predeterminado: private Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● private ● public
vgw_id	String	Especifica el ID del gateway virtual. Mínimo: 36 Máximo: 36
vlan	Integer	Especifica la VLAN para la conexión al gateway de usuario. El valor oscila entre 0 y 3999 . Mínimo: 0 Máximo: 3999
route_limit	Integer	Especifica las configuraciones de enrutamiento de subred remoto de la interfaz virtual. Mínimo: 1 Máximo: 200 Predeterminado: 50
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa Bidirectional Forwarding Detection (BFD). El valor puede ser true o false .
lag_id	String	Especifica el ID del LAG asociado a la interfaz virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece la interfaz virtual. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Parámetro	Tipo	Descripción
local_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
rate_limit	Boolean	Especifica si la limitación de velocidad está habilitada en una interfaz virtual.
address_family	String	Especifica la familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 . Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
local_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
lgw_id	String	Especifica el ID del gateway local, que se utiliza en escenarios de IES.
gateway_id	String	Especifica el ID del gateway asociado a la interfaz virtual.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.

Parámetro	Tipo	Descripción
priority	String	<p>Especifica la prioridad de una interfaz virtual. El valor puede ser normal o low. Si las prioridades son las mismas, las interfaces virtuales funcionan en modo de equilibrio de carga. Si las prioridades son diferentes, las interfaces virtuales funcionan en pares activo/en espera. El tráfico saliente se reenvía preferentemente a la interfaz virtual normal con una prioridad más alta. Esta opción solo es compatible con interfaces virtuales que utilizan enrutamiento de BGP.</p> <p>Predeterminado: normal</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● normal ● low
vif_peers	Array of VifPeer objects	Proporciona información acerca de los pares de la interfaz virtual.
extend_attribute	VifExtendAttribute object	Proporciona información de parámetros extendida.

Tabla 5-86 Etiqueta

Parámetro	Tipo	Descripción
key	String	<p>Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_).</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>
value	String	<p>Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.).</p> <p>Mínimo: 0</p> <p>Máximo: 43</p>

Tabla 5-87 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: ● bgp ● static
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.

Parámetro	Tipo	Descripción
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de la interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Especifica el estado del par de la interfaz virtual.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Tabla 5-88 VifExtendAttribute

Parámetro	Tipo	Descripción
ha_type	String	Especifica el tipo de detección de disponibilidad de la interfaz virtual. Valores de enumeración: <ul style="list-style-type: none"> ● nqa ● bfd
ha_mode	String	Especifica el modo de detección de disponibilidad. Valores de enumeración: <ul style="list-style-type: none"> ● auto_single ● auto_multi ● static_single ● static_multi ● enhance_nqa

Parámetro	Tipo	Descripción
detect_multiplier	Integer	Especifica el número de reintentos de detección. Predeterminado: 5
min_rx_interval	Integer	Especifica el intervalo para recibir paquetes de detección. Predeterminado: 1000
min_tx_interval	Integer	Especifica el intervalo para enviar paquetes de detección. Predeterminado: 1000
remote_disclaim	Integer	Especifica el identificador remoto de la sesión BFD estática.
local_disclaim	Integer	Especifica el identificador local de la sesión BFD estática.

Ejemplo de las solicitudes

Consultar una interfaz virtual

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

Ejemplo de las respuestas

Código de estado: 200

Aceptar

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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 huaweicloudskdc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskdc.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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.3.2 Actualización de una interfaz virtual

Función

Esta API se utiliza para actualizar una interfaz virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-89 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
virtual_interface_id	Sí	String	Especifica el ID de la interfaz virtual. Mínimo: 36 Máximo: 36

Parámetros de solicitud

Tabla 5-90 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-91 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
virtual_interface	Sí	UpdateVirtualInterface object	Objetos de interfaz virtual que se actualizarán

Tabla 5-92 UpdateVirtualInterface

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre de la interfaz virtual. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre la interfaz virtual. Mínimo: 0 Máximo: 128
bandwidth	No	Integer	Especifica la configuración de ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647
remote_ep_group	No	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	No	Array of strings	Especifica el grupo de punto de conexión local que está conectado a la red pública a través de una conexión.
enable_bfd	No	Boolean	Especifica si se activa BFD. El valor puede ser true o false .
enable_nqa	No	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
status	No	String	Confirma las interfaces virtuales creadas por otros usuarios. El valor puede ser ACCEPTED o REJECTED . Valores de enumeración: ● ACCEPTED ● REJECTED

Parámetros de respuesta

Código de estado: 200

Tabla 5-93 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
virtual_interface	VirtualInterface object	Especifica los parámetros para crear una interfaz virtual.

Tabla 5-94 VirtualInterface

Parámetro	Tipo	Descripción
id	String	Especifica el ID de la interfaz virtual. Máximo: 36
name	String	Especifica el nombre de la interfaz virtual. Máximo: 64
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false .
bandwidth	Integer	Especifica el ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647
create_time	String	Especifica la hora en que se creó la interfaz virtual. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ . Máximo: 255
description	String	Proporciona la información complementaria sobre la interfaz virtual. Máximo: 128
direct_connect_id	String	Especifica el ID de conexión. Máximo: 36
service_type	String	Especifica el tipo de gateway, que puede ser VGW , GDGW o LGW . Valores de enumeración: <ul style="list-style-type: none"> ● VGW ● GDGW ● LGW
status	String	Especifica el estado de funcionamiento, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE , PENDING_DELETE , DELETED , AUTHORIZATION o REJECTED .

Parámetro	Tipo	Descripción
tenant_id	String	Especifica el ID del proyecto. Mínimo: 32 Máximo: 32
type	String	Especifica el tipo de la interfaz virtual. El valor es private . Predeterminado: private Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● private ● public
vgw_id	String	Especifica el ID del gateway virtual. Mínimo: 36 Máximo: 36
vlan	Integer	Especifica la VLAN para la conexión al gateway de usuario. El valor oscila entre 0 y 3999 . Mínimo: 0 Máximo: 3999
route_limit	Integer	Especifica las configuraciones de enrutamiento de subred remoto de la interfaz virtual. Mínimo: 1 Máximo: 200 Predeterminado: 50
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa Bidirectional Forwarding Detection (BFD). El valor puede ser true o false .
lag_id	String	Especifica el ID del LAG asociado a la interfaz virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece la interfaz virtual. Mínimo: 36 Máximo: 36

Parámetro	Tipo	Descripción
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
local_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
rate_limit	Boolean	Especifica si la limitación de velocidad está habilitada en una interfaz virtual.
address_family	String	Especifica la familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 . Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
local_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
lgw_id	String	Especifica el ID del gateway local, que se utiliza en escenarios de IES.
gateway_id	String	Especifica el ID del gateway asociado a la interfaz virtual.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.

Parámetro	Tipo	Descripción
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
priority	String	Especifica la prioridad de una interfaz virtual. El valor puede ser normal o low . Si las prioridades son las mismas, las interfaces virtuales funcionan en modo de equilibrio de carga. Si las prioridades son diferentes, las interfaces virtuales funcionan en pares activo/en espera. El tráfico saliente se reenvía preferentemente a la interfaz virtual normal con una prioridad más alta. Esta opción solo es compatible con interfaces virtuales que utilizan enrutamiento de BGP. Predeterminado: normal Valores de enumeración: <ul style="list-style-type: none"> ● normal ● low
vif_peers	Array of VifPeer objects	Proporciona información acerca de los pares de la interfaz virtual.
extend_attribute	VifExtendAttribute object	Proporciona información de parámetros extendida.

Tabla 5-95 Tag

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Tabla 5-96 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de la interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: ● bgp ● static
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.

Parámetro	Tipo	Descripción
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Especifica el estado del par de la interfaz virtual.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Tabla 5-97 VifExtendAttribute

Parámetro	Tipo	Descripción
ha_type	String	Especifica el tipo de detección de disponibilidad de la interfaz virtual. Valores de enumeración: <ul style="list-style-type: none"> ● nqa ● bfd
ha_mode	String	Especifica el modo de detección de disponibilidad. Valores de enumeración: <ul style="list-style-type: none"> ● auto_single ● auto_multi ● static_single ● static_multi ● enhance_nqa

Parámetro	Tipo	Descripción
detect_multiplier	Integer	Especifica el número de reintentos de detección. Predeterminado: 5
min_rx_interval	Integer	Especifica el intervalo para recibir paquetes de detección. Predeterminado: 1000
min_tx_interval	Integer	Especifica el intervalo para enviar paquetes de detección. Predeterminado: 1000
remote_disclaim	Integer	Especifica el identificador remoto de la sesión BFD estática.
local_disclaim	Integer	Especifica el identificador local de la sesión BFD estática.

Ejemplo de solicitudes

Actualizar el nombre y la descripción de una interfaz virtual y cambiar su ancho de banda a 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
  }
}
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Actualizar el nombre y la descripción de una interfaz virtual y cambiar su ancho de banda a 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

Actualizar el nombre y la descripción de una interfaz virtual y cambiar su ancho de banda a 2 Mbit/s

```
# coding: utf-8

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

Actualizar el nombre y la descripción de una interfaz virtual y cambiar su ancho de banda a 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)  
    }  
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.3.3 Eliminación de una interfaz virtual

Función

Esta API se utiliza para eliminar una interfaz virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-98 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetro	Obligatorio	Tipo	Descripción
virtual_interfac e_id	Sí	String	Especifica el ID de la interfaz virtual. Mínimo: 36 Máximo: 36

Parámetros de solicitud

Tabla 5-99 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

No hay

Ejemplo de solicitudes

Eliminación de una interfaz virtual

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

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.3.4 Consulta de la lista de interfaces virtuales

Función

Esta API se utiliza para consultar todas las interfaces virtuales de un tenant.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-100 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-101 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000

Parámetro	Obligatorio	Tipo	Descripción
marker	No	String	Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit . Mínimo: 0 Máximo: 36
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5
sort_dir	No	Array	Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc .
sort_key	No	String	Especifica el campo de la ordenación. Predeterminado: id Mínimo: 0 Máximo: 36
enterprise_project_id	No	Array	Filtra instancias de recursos por ID de proyecto de empresa. Longitud del arreglo: 1 - 10
id	No	Array	Especifica el ID de recurso por el que se filtran las instancias. Longitud del arreglo: 1 - 5
status	No	Array	Especifica el estado por el que se filtran las instancias. Longitud del arreglo: 1 - 5
direct_connect_id	No	Array	Especifica el ID de conexión por el que se filtran y consultan las conexiones. Longitud del arreglo: 1 - 5
vgw_id	No	Array	Especifica el ID de gateway virtual por el que se filtran y consultan los gateway virtuales. Longitud del arreglo: 1 - 5

Parámetros de solicitud

Tabla 5-102 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-103 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
virtual_interfaces	Array of VirtualInterface objects	Especifica los parámetros para crear una interfaz virtual.
page_info	PageInfo object	Especifica la información de consulta de paginación.

Tabla 5-104 VirtualInterface

Parámetro	Tipo	Descripción
id	String	Especifica el ID de la interfaz virtual. Máximo: 36
name	String	Especifica el nombre de la interfaz virtual. Máximo: 64
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false .
bandwidth	Integer	Especifica el ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647

Parámetro	Tipo	Descripción
create_time	String	Especifica la hora en que se creó la interfaz virtual. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ . Máximo: 255
description	String	Proporciona la información complementaria sobre la interfaz virtual. Máximo: 128
direct_connect_id	String	Especifica el ID de conexión. Máximo: 36
service_type	String	Especifica el tipo de gateway, que puede ser VGW , GDGW o LGW . Valores de enumeración: <ul style="list-style-type: none"> ● VGW ● GDGW ● LGW
status	String	Especifica el estado de funcionamiento, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE , PENDING_DELETE , DELETED , AUTHORIZATION o REJECTED .
tenant_id	String	Especifica el ID del proyecto. Mínimo: 32 Máximo: 32
type	String	Especifica el tipo de la interfaz virtual. El valor es private . Predeterminado: private Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● private ● public
vgw_id	String	Especifica el ID del gateway virtual. Mínimo: 36 Máximo: 36
vlan	Integer	Especifica la VLAN para la conexión al gateway de usuario. El valor oscila entre 0 y 3999 . Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
route_limit	Integer	Especifica las configuraciones de enrutamiento de subred remoto de la interfaz virtual. Mínimo: 1 Máximo: 200 Predeterminado: 50
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa Bidirectional Forwarding Detection (BFD). El valor puede ser true o false .
lag_id	String	Especifica el ID del LAG asociado a la interfaz virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece la interfaz virtual. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
local_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
rate_limit	Boolean	Especifica si la limitación de velocidad está habilitada en una interfaz virtual.
address_family	String	Especifica la familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 . Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.

Parámetro	Tipo	Descripción
local_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
lgw_id	String	Especifica el ID del gateway local, que se utiliza en escenarios de IES.
gateway_id	String	Especifica el ID del gateway asociado a la interfaz virtual.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
bgp_route_limit	Integer	Especifica la configuración del enrutamiento de BGP.
priority	String	Especifica la prioridad de una interfaz virtual. El valor puede ser normal o low . Si las prioridades son las mismas, las interfaces virtuales funcionan en modo de equilibrio de carga. Si las prioridades son diferentes, las interfaces virtuales funcionan en pares activo/en espera. El tráfico saliente se reenvía preferentemente a la interfaz virtual normal con una prioridad más alta. Esta opción solo es compatible con interfaces virtuales que utilizan enrutamiento de BGP. Predeterminado: normal Valores de enumeración: <ul style="list-style-type: none"> ● normal ● low
vif_peers	Array of VifPeer objects	Proporciona información acerca de los pares de la interfaz virtual.
extend_attribute	VifExtendAttribute object	Proporciona información de parámetros extendida.

Tabla 5-105 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Tabla 5-106 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de la interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.

Parámetro	Tipo	Descripción
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● bgp ● static
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración del enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Especifica el estado del par de la interfaz virtual.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Tabla 5-107 VifExtendAttribute

Parámetro	Tipo	Descripción
ha_type	String	<p>Especifica el tipo de detección de disponibilidad de la interfaz virtual.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● nqa ● bfd
ha_mode	String	<p>Especifica el modo de detección de disponibilidad.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● auto_single ● auto_multi ● static_single ● static_multi ● enhance_nqa
detect_multiplier	Integer	<p>Especifica el número de reintentos de detección.</p> <p>Predeterminado: 5</p>
min_rx_interval	Integer	<p>Especifica el intervalo para recibir paquetes de detección.</p> <p>Predeterminado: 1000</p>
min_tx_interval	Integer	<p>Especifica el intervalo para enviar paquetes de detección.</p> <p>Predeterminado: 1000</p>
remote_disclaim	Integer	<p>Especifica el identificador remoto de la sesión BFD estática.</p>
local_disclaim	Integer	<p>Especifica el identificador local de la sesión BFD estática.</p>

Tabla 5-108 PageInfo

Parámetro	Tipo	Descripción
previous_marker	String	<p>Especifica el marcador de la página anterior. El valor es el UUID del recurso.</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>
current_count	Integer	<p>Especifica el número de recursos de la lista actual.</p> <p>Mínimo: 0</p> <p>Máximo: 2000</p>

Parámetro	Tipo	Descripción
next_marker	String	<p>Especifica el marcador de la página siguiente. El valor es el UUID del recurso. Si el valor está vacío, el recurso se encuentra en la última página.</p> <p>Mínimo: 0 Máximo: 36</p>

Ejemplo de solicitudes

Consulta de la lista de interfaces virtuales

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

Ejemplo de respuestas

Código de estado: 200

Aceptar

```
{
  "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"  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.3.5 Creación de una interfaz virtual

Función

Esta API se utiliza para crear una interfaz virtual.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-109 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-110 Parámetros de cabecera de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-111 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
virtual_interfac e	Sí	CreateVirtualI nterface object	Especifica los parámetros para crear una interfaz virtual.

Tabla 5-112 CreateVirtualInterface

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre de la interfaz virtual. Mínimo: 0 Máximo: 64
description	No	String	Proporciona la información complementaria sobre la interfaz virtual. Mínimo: 0 Máximo: 128
direct_connect_id	No	String	Especifica el ID de la conexión asociada a la interfaz virtual. Al crear una interfaz virtual, debe especificar direct_connect_id o lag_id . Este parámetro es obligatorio cuando LAG no se admite en el sitio.
type	Sí	String	Especifica el tipo de la interfaz virtual. El valor es private . Valores de enumeración: ● private ● public
service_type	No	String	Especifica el tipo de gateway, que puede ser VGW , GDBGW o LGW . Valores de enumeración: ● VGW ● GDBGW ● LGW
vlan	Sí	Integer	Especifica la VLAN del cliente que se va a conectar. Si selecciona una conexión alojada, la VLAN debe ser la misma que la de la conexión alojada. Mínimo: 0 Máximo: 3999
bandwidth	Sí	Integer	Especifica el ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647

Parámetro	Obligatorio	Tipo	Descripción
local_gateway_v4_ip	No	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado en la nube. Este parámetro es obligatorio si address_family está establecido en una dirección IPv4.
remote_gateway_v4_ip	No	String	Especifica la dirección de la interfaz IPv4 del gateway en la red local. Este parámetro es obligatorio si address_family está establecido en una dirección IPv4.
address_family	No	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 . Predeterminado: ipv4
local_gateway_v6_ip	No	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado en la nube. Este parámetro es obligatorio si address_family está establecido en una dirección IPv6.
remote_gateway_v6_ip	No	String	Especifica la dirección de la interfaz IPv6 del gateway en la red local. Este parámetro es obligatorio si address_family está establecido en una dirección IPv6.
vgw_id	Sí	String	Especifica el ID del gateway virtual conectado por la interfaz virtual.
route_mode	Sí	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Predeterminado: static Valores de enumeración: ● static ● bgp
bgp_asn	No	Integer	Especifica el ASN del par BGP en el lado del cliente. Mínimo: 1 Máximo: 4294967295
bgp_md5	No	String	Especifica la contraseña de MD5 del par BGP.

Parámetro	Obligatorio	Tipo	Descripción
remote_ep_group	Sí	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	No	Array of strings	Especifica las subredes que tienen acceso a los servicios de Internet a través de una conexión directa.
enable_bfd	No	Boolean	Especifica si se activa Bidirectional Forwarding Detection (BFD). El valor puede ser true o false . Predeterminado: false
enable_nqa	No	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false . Default: false
lag_id	No	String	Especifica el ID del LAG asociado a la interfaz virtual. Mínimo: 36 Máximo: 36
resource_tenant_id	No	String	Especifica el ID de proyecto de otro tenant, que se utiliza para crear las interfaces virtuales entre tenants.
enterprise_project_id	No	String	Especifica el ID del proyecto de empresa al que pertenece el recurso. Mínimo: 36 Máximo: 36
tags	No	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10

Tabla 5-113 Etiqueta

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	No	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Parámetros de respuesta

Código de estado: 201

Tabla 5-114 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
virtual_interface	VirtualInterface object	Especifica los parámetros para crear una interfaz virtual.

Tabla 5-115 VirtualInterface

Parámetro	Tipo	Descripción
id	String	Especifica el ID de la interfaz virtual. Máximo: 36
name	String	Especifica el nombre de la interfaz virtual. Máximo: 64
admin_state_up	Boolean	Especifica el estado administrativo, que puede ser true o false .
bandwidth	Integer	Especifica el ancho de banda de la interfaz virtual. Mínimo: 2 Máximo: 2147483647

Parámetro	Tipo	Descripción
create_time	String	Especifica la hora en que se creó la interfaz virtual. Se utiliza el formato de hora UTC aaaa-MM-ddTHH:mm:ss.SSSZ . Máximo: 255
description	String	Proporciona la información complementaria sobre la interfaz virtual. Máximo: 128
direct_connect_id	String	Especifica el ID de conexión. Máximo: 36
service_type	String	Especifica el tipo de gateway, que puede ser VGW , GDGW o LGW . Valores de enumeración: <ul style="list-style-type: none"> ● VGW ● GDGW ● LGW
status	String	Especifica el estado de funcionamiento, que puede ser ACTIVE , DOWN , BUILD , ERROR , PENDING_CREATE , PENDING_UPDATE , PENDING_DELETE , DELETED , AUTHORIZATION o REJECTED .
tenant_id	String	Especifica el ID del proyecto. Mínimo: 32 Máximo: 32
type	String	Especifica el tipo de la interfaz virtual. El valor es private . Predeterminado: private Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● private ● public
vgw_id	String	Especifica el ID del gateway virtual. Mínimo: 36 Máximo: 36
vlan	Integer	Especifica la VLAN para la conexión al gateway de usuario. El valor oscila entre 0 y 3999 . Mínimo: 0 Máximo: 3999

Parámetro	Tipo	Descripción
route_limit	Integer	Especifica las configuraciones de enrutamiento de subred remoto de la interfaz virtual. Mínimo: 1 Máximo: 200 Predeterminado: 50
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa Bidirectional Forwarding Detection (BFD). El valor puede ser true o false .
lag_id	String	Especifica el ID del LAG asociado a la interfaz virtual. Mínimo: 36 Máximo: 36
device_id	String	Especifica el ID del dispositivo al que pertenece la interfaz virtual.
enterprise_project_id	String	Especifica el ID del proyecto de empresa al que pertenece la interfaz virtual. Mínimo: 36 Máximo: 36
tags	Array of Tag objects	Especifica las etiquetas. Longitud del arreglo: 0 - 10
local_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v4_ip	String	Especifica la dirección de la interfaz IPv4 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
ies_id	String	Especifica el identificador de un emplazamiento de borde de IES.
reason	String	Muestra información de error si el estado de una línea es Error .
rate_limit	Boolean	Especifica si la limitación de velocidad está habilitada en una interfaz virtual.
address_family	String	Especifica la familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 . Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.

Parámetro	Tipo	Descripción
local_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado en la nube. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
remote_gateway_v6_ip	String	Especifica la dirección de la interfaz IPv6 del gateway utilizado in situ. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
lgw_id	String	Especifica el ID del gateway local, que se utiliza en escenarios de IES.
gateway_id	String	Especifica el ID del gateway asociado a la interfaz virtual.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública. Este parámetro se ha migrado a la lista de parámetros vifpeer y se descartará más adelante.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
priority	String	Especifica la prioridad de una interfaz virtual. El valor puede ser normal o low . Si las prioridades son las mismas, las interfaces virtuales funcionan en modo de equilibrio de carga. Si las prioridades son diferentes, las interfaces virtuales funcionan en pares activo/en espera. El tráfico saliente se reenvía preferentemente a la interfaz virtual normal con una prioridad más alta. Esta opción solo es compatible con interfaces virtuales que utilizan enrutamiento de BGP. Predeterminado: normal Valores de enumeración: <ul style="list-style-type: none"> ● normal ● low
vif_peers	Array of VifPeer objects	Proporciona información acerca de los pares de la interfaz virtual.
extend_attribute	VifExtendAttribute object	Proporciona información de parámetros extendida.

Tabla 5-116 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Tabla 5-117 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de la interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.

Parámetro	Tipo	Descripción
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● bgp ● static
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Specifies the status of the virtual interface peer.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Tabla 5-118 VifExtendAttribute

Parámetro	Tipo	Descripción
ha_type	String	Especifica el tipo de detección de disponibilidad de la interfaz virtual. Valores de enumeración: <ul style="list-style-type: none"> ● nqa ● bfd
ha_mode	String	Especifica el modo de detección de disponibilidad. Valores de enumeración: <ul style="list-style-type: none"> ● auto_single ● auto_multi ● static_single ● static_multi ● enhance_nqa
detect_multiplier	Integer	Especifica el número de reintentos de detección. Predeterminado: 5
min_rx_interval	Integer	Especifica el intervalo para recibir paquetes de detección. Predeterminado: 1000
min_tx_interval	Integer	Especifica el intervalo para enviar paquetes de detección. Predeterminado: 1000
remote_disclaim	Integer	Especifica el identificador remoto de la sesión BFD estática.
local_disclaim	Integer	Especifica el identificador local de la sesión BFD estática.

Ejemplo de las solicitudes

Creación de una interfaz virtual privada, con el ancho de banda establecido a 2 Mbit/s, VLAN a 332, gateway local a 1.1.1.1/30, y gateway remoto a 1.1.1.2/30, y modo de enrutamiento a enrutamiento estático

```
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" ]
}
}
```

Ejemplo de las respuestas

Código de estado: 201

Creado

```
{
  "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"
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Creación de una interfaz virtual privada, con el ancho de banda establecido a 2 Mbit/s, VLAN a 332, gateway local a 1.1.1.1/30, y gateway remoto a 1.1.1.2/30, y modo de enrutamiento a enrutamiento estático

```
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

Creación de una interfaz virtual privada, con el ancho de banda establecido a 2 Mbit/s, VLAN a 332, gateway local a 1.1.1.1/30, y gateway remoto a 1.1.1.2/30, y modo de enrutamiento a enrutamiento estático

```
# 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

Creación de una interfaz virtual privada, con el ancho de banda establecido a 2 Mbit/s, VLAN a 332, gateway local a 1.1.1.1/30, y gateway remoto a 1.1.1.2/30, y modo de enrutamiento a enrutamiento estático

```
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)
}
}

```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
201	Creado

Códigos de error

Consulte [Códigos de error](#).

5.3.6 Actualización de un par de interfaz virtual

Función

Esta API se utiliza para actualizar un par de la interfaz virtual, por ejemplo, su subred remota, nombre y descripción. Esta API solo está disponible en regiones que admiten IPv6. Para utilizar esta API, póngase en contacto con el servicio de atención al cliente.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-119 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
vif_peer_id	Sí	String	Especifica el identificador del par de la interfaz virtual.

Parámetros de solicitud

Tabla 5-120 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	<p>Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i>. El token es el valor de X-Subject-Token en el encabezado de respuesta.</p> <p>Mínimo: 0 Máximo: 10240</p>

Tabla 5-121 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
vif_peer	No	UpdateVifPeer object	Especifica los parámetros para actualizar un par de interfaz virtual.

Tabla 5-122 UpdateVifPeer

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	<p>Especifica el nombre del par de interfaz virtual.</p> <p>Mínimo: 0 Máximo: 64</p>
description	No	String	<p>Proporciona información adicional sobre el par de la interfaz virtual.</p> <p>Mínimo: 0 Máximo: 128</p>
remote_ep_group	No	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.

Parámetros de respuesta

Código de estado: 200

Tabla 5-123 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
vif_peer	VifPeer object	Especifica el par de la interfaz virtual.

Tabla 5-124 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de la interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● bgp ● static

Parámetro	Tipo	Descripción
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Especifica el estado del par de la interfaz virtual.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Especifica si se va a habilitar NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Ejemplo de solicitudes

Actualizar el nombre, la descripción y la subred remota de un par de interfaz virtual

```
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" ]  
}  
}
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

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

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Actualizar el nombre, la descripción y la subred remota de un par de interfaz virtual

```
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

Actualizar el nombre, la descripción y la subred remota de un par de interfaz virtual

```
# coding: utf-8

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

Actualizar el nombre, la descripción y la subred remota de un par de interfaz virtual

```
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)
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.3.7 Eliminación de un par de interfaz virtual

Función

Esta API se utiliza para eliminar una interfaz virtual. La interfaz virtual debe contener al menos un par. No se puede eliminar el último par. Esta API solo está disponible en regiones que admiten IPv6. Para utilizar esta API, póngase en contacto con el servicio de atención al cliente.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-125 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
vif_peer_id	Sí	String	Especifica el identificador del par de la interfaz virtual.

Parámetros de solicitud

Tabla 5-126 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

No hay

Ejemplo de solicitudes

Eliminación de un par de interfaz virtual

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

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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 huaweicloudskdkc.v3.region.dc_region import DcRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudskdkc.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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.3.8 Creación de un par de interfaz virtual

Función

Esta API se utiliza para crear un par IPv6. Cada interfaz virtual puede tener dos pares: los pares IPv4 e IPv6. Cuando se crea una interfaz virtual, se crea un par IPv4 de forma predeterminada. Después de crear un par de la interfaz virtual, puede consultar su configuración mediante la visualización de los detalles de la interfaz virtual. Esta API solo está disponible en regiones que admiten IPv6. Para utilizar esta API, póngase en contacto con el servicio de atención al cliente.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-127 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-128 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-129 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
vif_peer	No	CreateVifPeer object	Especifica los parámetros para crear un par de la interfaz virtual.

Tabla 5-130 CreateVifPeer

Parámetro	Obligatorio	Tipo	Descripción
name	No	String	Especifica el nombre del par de la interfaz virtual. Mínimo: 0 Máximo: 64
description	No	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	No	String	Especifica la familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	No	String	Especifica la dirección IP del gateway del lado de la nube del par de la interfaz virtual.
remote_gateway_ip	No	String	Especifica la dirección de gateway del cliente del par de la interfaz virtual.
route_mode	No	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: ● bgp ● static
bgp_asn	No	Integer	Especifica el AS del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	No	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	No	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.

Parámetro	Obligatorio	Tipo	Descripción
vif_id	No	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36

Parámetros de respuesta

Código de estado: 201

Tabla 5-131 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
vif_peer	VifPeer object	Especifica el par de la interfaz virtual.

Tabla 5-132 VifPeer

Parámetro	Tipo	Descripción
id	String	Especifica el ID del recurso. Mínimo: 36 Máximo: 36
tenant_id	String	Especifica el ID del proyecto al que pertenece el par de interfaz virtual. Mínimo: 36 Máximo: 36
name	String	Especifica el nombre del par de la interfaz virtual. Mínimo: 0 Máximo: 64
description	String	Proporciona información adicional sobre el par de la interfaz virtual. Mínimo: 0 Máximo: 128
address_family	String	Especifica el tipo de familia de direcciones de la interfaz virtual, que puede ser IPv4 o IPv6 .
local_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en la nube.
remote_gateway_ip	String	Especifica la dirección del par de interfaz virtual utilizado en el centro de datos local.

Parámetro	Tipo	Descripción
route_mode	String	Especifica el modo de enrutamiento, que puede ser static o bgp . Máximo: 255 Valores de enumeración: <ul style="list-style-type: none"> ● bgp ● static
bgp_asn	Integer	Especifica el ASN del par de BGP. Mínimo: 1 Máximo: 4294967295
bgp_md5	String	Especifica la contraseña de MD5 del par BGP.
remote_ep_group	Array of strings	Especifica la lista de subred remota, que registra los bloques CIDR utilizados en el centro de datos local.
service_ep_group	Array of strings	Especifica la lista de direcciones de red pública a las que puede acceder el centro de datos local. Este campo es obligatorio en las API de conexiones de red pública.
device_id	String	Especifica el ID del dispositivo al que pertenece el par de interfaz virtual.
bgp_route_limit	Integer	Especifica la configuración de enrutamiento de BGP.
bgp_status	String	Especifica el estado del protocolo BGP del par de interfaz virtual. Si el par de la interfaz virtual utiliza enrutamiento estático, el estado es null . Máximo: 10
status	String	Especifica el estado del par de la interfaz virtual.
vif_id	String	Especifica el ID de la interfaz virtual correspondiente al par de la interfaz virtual. Mínimo: 36 Máximo: 36
receive_route_number	Integer	Especifica el número de enrutamientos de BGP recibidos si se utiliza el enrutamiento de BGP. Si se utiliza el enrutamiento estático, este parámetro no tiene sentido y el valor es -1 . Nota: Si no se puede obtener este parámetro, póngase en contacto con el servicio de atención al cliente para migrar los puertos.
enable_nqa	Boolean	Specifies whether to enable NQA. El valor puede ser true o false .
enable_bfd	Boolean	Especifica si se activa BFD. El valor puede ser true o false .

Ejemplo de solicitudes

Creación de un par de la interfaz virtual con la subred remota establecida en 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" ]
  }
}
```

Ejemplo de respuestas

Código de estado: 201

Creado

```
{
  "vif_peer" : {
    "name" : "vif-0819",
    "description" : "mytest",
    "id" : "4c95de3e-9f75-4357-9c79-b22498dd71c7",
    "tenant_id" : "ed28c294165741faeaccab26913122a1",
    "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" ]
  }
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Creación de un par de la interfaz virtual con la subred remota establecida en 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

Creación de un par de la interfaz virtual con la subred remota establecida en 20.1.1.0/24

```
# coding: utf-8

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

Creación de un par de la interfaz virtual con la subred remota establecida en 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.

Códigos de estado

Código de estado	Descripción
201	Creado

Códigos de error

Consulte [Códigos de error](#).

5.3.9 Realización de una prueba de conmutación de interfaz virtual

Función

Si hay dos conexiones, se requiere la conmutación automática entre las conexiones para las pruebas de conectividad. Si se realiza una conmutación, la interfaz virtual de la conexión en uso está deshabilitada y el tráfico se interrumpe. Puede realizar dos operaciones en una interfaz virtual:- Ejecutar el comando **shutdown** para deshabilitar la interfaz virtual.- Ejecutar el comando **undo_shutdown** para habilitar la interfaz virtual. Si se selecciona **shutdown** para la prueba de conmutación, la interfaz virtual está en el estado **ADMIN_SHUTDOWN** y no se pueden realizar operaciones en la interfaz virtual. Si se selecciona **undo_shutdown** para la prueba de conmutación, la interfaz virtual está en el estado **ACTIVE**.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-133 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Parámetros de solicitud

Tabla 5-134 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Tabla 5-135 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
switchover_test_record	No	CreateSwitchoverTest object	Especifica los parámetros para crear un registro de prueba de conmutación.

Tabla 5-136 CreateSwitchoverTest

Parámetro	Obligatorio	Tipo	Descripción
resource_id	Sí	String	Especifica el ID del recurso en el que se va a realizar la prueba de conmutación.

Parámetro	Obligatorio	Tipo	Descripción
resource_type	No	String	Especifica el tipo del recurso en el que se va a realizar la prueba de conmutación. Predeterminado: virtual_interface Mínimo: 0 Máximo: 128 Valores de enumeración: ● virtual_interface
operation	Sí	String	Especifica si se debe realizar una prueba de conmutación. Opciones de valor: shutdown y undo_shutdown Mínimo: 0 Máximo: 128 Valores de enumeración: ● shutdown ● undo_shutdown

Parámetros de respuesta

Código de estado: 201

Tabla 5-137 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
switchover_test_record	SwitchoverTestRecord object	Proporciona información adicional sobre el registro de prueba de conmutación.

Tabla 5-138 SwitchoverTestRecord

Parámetro	Tipo	Descripción
id	String	Especifica el ID único del registro de prueba de conmutación.
tenant_id	String	Especifica el ID del tenant. Mínimo: 32 Máximo: 32

Parámetro	Tipo	Descripción
resource_id	String	Especifica el ID del recurso en el que se va a realizar la prueba de conmutación.
resource_type	String	Especifica el tipo del recurso en el que se va a realizar la prueba de conmutación. Predeterminado: virtual_interface Mínimo: 0 Máximo: 128 Valores de enumeración: ● virtual_interface
operation	String	Especifica si se debe realizar una prueba de conmutación. Opciones de valor: shutdown y undo_shutdown Mínimo: 0 Máximo: 128 Valores de enumeración: ● shutdown ● undo_shutdown
start_time	String	Especifica la hora de inicio de la prueba de conmutación.
end_time	String	Especifica la hora de finalización de la prueba de conmutación.
operate_status	String	Especifica el estado de la prueba de conmutación. ● STARTING : estado inicial ● INPROGRESS : entregando la configuración ● COMPLETE : configuración entregada ● ERROR Error en la entrega de la configuración Valores de enumeración: ● STARTING ● INPROGRESS ● COMPLETE ● ERROR

Ejemplo de solicitudes

Apagar la interfaz virtual para realizar una prueba de conmutación

```
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"  
}  
}
```

Ejemplo de respuestas

Código de estado: 201

Creado

```
{  
  "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"  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Apagar la interfaz virtual para realizar una prueba de conmutación

```
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();
        CreateSwitchoverTestRecordbody = new
CreateSwitchoverTestRecordbody();
        switchoverTestRecordbody.withResourceId("d0b3329c-0063-470c-
b1dc-657656b2e540")
        .withResourceType(CreateSwitchoverTest.ResourceTypeEnum.fromValue("vir
tual_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

Apagar la interfaz virtual para realizar una prueba de conmutación

```
# coding: utf-8

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

Apagar la interfaz virtual para realizar una prueba de conmutación

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

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
201	Creado

Códigos de error

Consulte [Códigos de error](#).

5.3.10 Consulta de los registros de prueba de conmutación de una interfaz virtual

Función

Esta API se utiliza para consultar la lista de registros de prueba de conmutación. Solo se muestran los registros cuyo `operate_status` sea **COMPELTE**.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

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

Tabla 5-139 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-140 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
limit	No	Integer	Especifica el número de registros devueltos en cada página. Rango de valores: 1 a 2000 Mínimo: 1 Máximo: 2000 Predeterminado: 2000

Parámetro	Obligatorio	Tipo	Descripción
marker	No	String	Especifica el ID del último registro de recursos de la página anterior. Si este parámetro se deja en blanco, se consulta la primera página. Este parámetro debe usarse junto con limit . Mínimo: 0 Máximo: 36
fields	No	Array	Especifica la lista de campos que se mostrarán. Longitud del arreglo: 1 - 5
sort_dir	No	Array	Especifica el orden de clasificación de los resultados devueltos, que puede ser asc (orden ascendente) o desc (orden descendente). El valor predeterminado es asc .
sort_key	No	String	Especifica el campo de la ordenación. Predeterminado: id Mínimo: 0 Máximo: 36
resource_id	No	Array	ID de recurso utilizado para consultar los registros de prueba de conmutación.

Parámetros de solicitud

Tabla 5-141 Parámetros de encabezado de solicitud

Parámetro	Obligatorio	Tipo	Descripción
X-Auth-Token	Sí	String	Especifica el token de usuario. Para obtener el token, consulte la sección "Obtención del token de usuario" en la <i>Referencia de las API de Identity and Access Management</i> . El token es el valor de X-Subject-Token en el encabezado de respuesta. Mínimo: 0 Máximo: 10240

Parámetros de respuesta

Código de estado: 200

Tabla 5-142 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
request_id	String	Especifica el ID de la solicitud.
switchover_test_records	Array of SwitchoverTestRecord objects	Muestra los registros de prueba de conmutación.
page_info	PageInfo object	Especifica la información de consulta de paginación.

Tabla 5-143 SwitchoverTestRecord

Parámetro	Tipo	Descripción
id	String	Especifica el ID único del registro de prueba de conmutación.
tenant_id	String	Especifica el ID del tenant. Mínimo: 32 Máximo: 32
resource_id	String	Especifica el ID del recurso en el que se va a realizar la prueba de conmutación.
resource_type	String	Especifica el tipo del recurso en el que se va a realizar la prueba de conmutación. Predeterminado: virtual_interface Mínimo: 0 Máximo: 128 Valores de enumeración: ● virtual_interface
operation	String	Especifica si se debe realizar una prueba de conmutación. Opciones de valor: shutdown y undo_shutdown Mínimo: 0 Máximo: 128 Valores de enumeración: ● shutdown ● undo_shutdown
start_time	String	Especifica la hora de inicio de la prueba de conmutación.

Parámetro	Tipo	Descripción
end_time	String	Especifica la hora de finalización de la prueba de conmutación.
operate_status	String	<p>Especifica el estado de la prueba de conmutación.</p> <ul style="list-style-type: none"> ● STARTING: estado inicial ● INPROGRESS: entregando la configuración ● COMPLETE: configuración entregada ● ERROR Error en la entrega de la configuración <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● STARTING ● INPROGRESS ● COMPLETE ● ERROR

Tabla 5-144 PageInfo

Parámetro	Tipo	Descripción
previous_marker	String	<p>Especifica el marcador de la página anterior. El valor es el UUID del recurso.</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>
current_count	Integer	<p>Especifica el número de recursos de la lista actual.</p> <p>Mínimo: 0</p> <p>Máximo: 2000</p>
next_marker	String	<p>Especifica el marcador de la página siguiente. El valor es el UUID del recurso. Si el valor está vacío, el recurso se encuentra en la última página.</p> <p>Mínimo: 0</p> <p>Máximo: 36</p>

Ejemplo de solicitudes

Consulta de la lista de registros de prueba de conmutación

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

Ejemplo de respuestas

Código de estado: 200

Aceptar

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

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
}
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.4 Gestión de etiquetas

5.4.1 Consulta de etiquetas por tipo de recurso

Función

Esta API se utiliza para consultar las etiquetas de un tipo de recurso en un proyecto específico. TMS utiliza esta API para enumerar las etiquetas creadas por un tenant para facilitar la creación de etiquetas y el filtrado de recursos en la consola.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

GET /v3/{project_id}/{resource_type}/tags

Tabla 5-145 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
resource_type	Sí	String	Especifica el tipo de recurso de Direct Connect. <ul style="list-style-type: none"> ● dc-directconnect: conexión ● dc-vgw: gateway virtual ● dc-vif: interfaz virtual Valores de enumeración: <ul style="list-style-type: none"> ● dc-directconnect ● dc-vgw ● dc-vif

Parámetros de solicitud

No hay

Parámetros de respuesta

Código de estado: 200

Tabla 5-146 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
tags	Array of Tag objects	Especifica las etiquetas.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-147 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Ejemplo de solicitudes

```
GET https://{dc_endpoint}/v3/ed28c294165741faeaccab26913122a1/dc-directconnect/  
tags
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

```
{  
  "tags" : [ {  
    "key" : "department",  
    "value" : "finance"  
  } ]  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	OK

Códigos de error

Consulte [Códigos de error](#).

5.4.2 Consulta de etiquetas de recurso

Función

Esta API se utiliza para consultar las etiquetas de recursos.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

GET /v3/{project_id}/{resource_type}/{resource_id}/tags

Tabla 5-148 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
resource_type	Sí	String	Especifica el tipo de recurso de Direct Connect. <ul style="list-style-type: none"> ● dc-directconnect: conexión ● dc-vgw: gateway virtual ● dc-vif: interfaz virtual Valores de enumeración: <ul style="list-style-type: none"> ● dc-directconnect ● dc-vgw ● dc-vif
resource_id	Sí	String	Especifica el ID del recurso.

Parámetros de solicitud

No hay

Parámetros de respuesta

Código de estado: 200

Tabla 5-149 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
tags	Array of Tag objects	Especifica la lista de etiquetas.

Parámetro	Tipo	Descripción
sys_tags	Array of Tag objects	Especifica la lista de etiquetas consultadas. Si no hay ninguna etiqueta coincidente, se devuelve una matriz vacía.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-150 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Ejemplo de solicitudes

Consulta de etiquetas de un recurso

```
https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/resource-instances/action
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

```
{
  "request_id" : "80ef5f21-b81a-4546-b23d-84272507d330",
  "tags" : [ {
    "key" : "key2",
    "value" : "value2"
  }, {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key3",
    "value" : "value3"
  } ]
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.4.3 Adición de una etiqueta de recurso

Función

- Un recurso puede tener un máximo de 10 etiquetas.
- Esta API es idempotente.
- Si una etiqueta que va a crear tiene la misma clave que una etiqueta existente, la etiqueta se creará y sobrescribirá la existente.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

POST /v3/{project_id}/{resource_type}/{resource_id}/tags

Tabla 5-151 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
resource_id	Sí	String	Especifica el ID del recurso.

Parámetro	Obligatorio	Tipo	Descripción
resource_type	Sí	String	<p>Especifica el tipo de recurso de Direct Connect.</p> <ul style="list-style-type: none"> ● dc-directconnect: conexión ● dc-vgw: gateway virtual ● dc-vif: interfaz virtual <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● dc-directconnect ● dc-vgw ● dc-vif

Parámetros de solicitud

Tabla 5-152 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
tag	Sí	Tag object	Especifica la etiqueta de recurso.

Tabla 5-153 Etiqueta

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	<p>Specifies the tag key. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_).</p> <p>Mínimo: 0 Máximo: 36</p>
value	No	String	<p>Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.).</p> <p>Mínimo: 0 Máximo: 43</p>

Parámetros de respuesta

No hay

Ejemplo de solicitudes

Agregar etiquetas a un recurso

```
POST https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/3320166e-
b937-40cc-a35c-02cd3f2b3ee2/tags

{
  "tag" : {
    "key" : "key1",
    "value" : "value1"
  }
}
```

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

Agregar etiquetas a un recurso

```
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

Agregar etiquetas a un recurso

```
# 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

Agregar etiquetas a un recurso

```
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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.4.4 Adición o eliminación de etiquetas por lotes

Función

Esta API se utiliza para agregar etiquetas por lotes a o eliminar etiquetas de un recurso especificado. TMS necesita usar esta API para gestionar etiquetas de recursos por lotes. Un recurso puede tener un máximo de 10 etiquetas.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

POST /v3/{project_id}/{resource_type}/{resource_id}/tags/action

Tabla 5-154 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
resource_id	Sí	String	Especifica el ID del recurso.
resource_type	Sí	String	Especifica el tipo de recurso de Direct Connect. <ul style="list-style-type: none"> ● dc-directconnect: conexión ● dc-vgw: gateway virtual ● dc-vif: interfaz virtual Valores de enumeración: <ul style="list-style-type: none"> ● dc-directconnect ● dc-vgw ● dc-vif

Parámetros de solicitud

Tabla 5-155 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
action	Sí	String	Especifica la operación. Las opciones son las siguientes: <ul style="list-style-type: none"> ● create ● delete Valores de enumeración: <ul style="list-style-type: none"> ● create ● delete
tags	No	Array of Tag objects	Especifica las etiquetas.
sys_tags	No	Array of Tag objects	Especifica las etiquetas del sistema.

Tabla 5-156 Etiqueta

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	No	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Parámetros de respuesta

No hay

Ejemplo de solicitudes

- Adición por lotes de las etiquetas a los recursos

```
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"
  } ]
}
```

- **Eliminación por lotes de las etiquetas de recursos**

POST <https://{endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/dc-vgw/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/action>

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

Ejemplo de respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

- **Adición por lotes de las etiquetas a los recursos**

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

- **Eliminación por lotes de las etiquetas de recursos**

```
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

- Adición por lotes de las etiquetas a los recursos

```
# 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)
```

- **Eliminación por lotes de las etiquetas de recursos**

```
# 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

- Adición por lotes de las etiquetas a los recursos

```
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)
        }
    }
}
```

- Eliminación por lotes de las etiquetas de recursos

```
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)
        }
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.4.5 Eliminación de una etiqueta de recurso

Función

Cuando se elimina una etiqueta, el conjunto de caracteres de etiqueta no se verifica. Antes de invocar a esta API, el URL de la API debe estar codificada. Si la clave que se va a suprimir no existe, se visualiza 404. La clave no se puede dejar en blanco ni una cadena vacía.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

DELETE /v3/{project_id}/{resource_type}/{resource_id}/tags/{key}

Tabla 5-157 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	Especifica la clave de etiqueta.
project_id	Sí	String	Especifica el ID del proyecto.
resource_id	Sí	String	Especifica el ID del recurso.
resource_type	Sí	String	Especifica el tipo de recurso de Direct Connect. <ul style="list-style-type: none">● dc-directconnect: conexión● dc-vgw: gateway virtual● dc-vif: interfaz virtual Valores de enumeración: <ul style="list-style-type: none">● dc-directconnect● dc-vgw● dc-vif

Parámetros de solicitud

No hay

Parámetros de respuesta

No hay

Ejemplo de las solicitudes

Eliminar etiquetas de un recurso

```
DELETE https://{dc-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/  
3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/key1
```

Ejemplo de las respuestas

No hay

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
}
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
204	Sin contenido

Códigos de error

Consulte [Códigos de error](#).

5.4.6 Consulta de recursos por etiqueta

Función

Esta API se utiliza para consultar recursos como los gateway virtuales y las interfaces virtuales por etiqueta.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

POST /v3/{project_id}/{resource_type}/resource-instances/action

Tabla 5-158 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.
resource_type	Sí	String	Especifica el tipo de recurso de Direct Connect. <ul style="list-style-type: none"> ● dc-directconnect: conexión ● dc-vgw: gateway virtual ● dc-vif: interfaz virtual Valores de enumeración: <ul style="list-style-type: none"> ● dc-directconnect ● dc-vgw ● dc-vif

Parámetros de solicitud

Tabla 5-159 Parámetros del cuerpo de solicitud

Parámetro	Obligatorio	Tipo	Descripción
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. Este parámetro no está disponible cuando action está establecida en count . Si action se establece en filter , el valor debe ser un número y el valor predeterminado es 0 . El valor no puede ser un número negativo.
limit	No	String	Especifica el número de registros que se van a consultar. Este parámetro no está disponible cuando action está establecida en count . Si action se establece en filter , el valor predeterminado es 1000 . El valor máximo es de 1000 y el valor mínimo de 1 . El valor no puede ser un número negativo.

Parámetro	Obligatorio	Tipo	Descripción
action	Sí	String	<p>Especifica la operación que se va a realizar, que solo puede ser filter (filtrar) o count (consultar el número total). filter indica la consulta de paginación. count indica que se devolverá el número total de resultados de consulta que cumplan los criterios de búsqueda. No se permite devolver otros campos.</p> <p>Valores de enumeración:</p> <ul style="list-style-type: none"> ● filter ● count
matches	No	Array of Match objects	<p>Especifica el criterio de búsqueda. key es el campo que corresponde, por ejemplo, a resource_name. value indica el valor que se va a hacer coincidir. Este campo es un valor de diccionario fijo.</p> <p>Determine si la coincidencia difusa es necesaria en función de diferentes campos. Por ejemplo, si key es resource_name, se utiliza la búsqueda difusa (sin distinción entre mayúsculas y minúsculas) de forma predeterminada. Si value es una cadena vacía, se utiliza la coincidencia exacta. Si key es resource_id, se utiliza la coincidencia exacta. Actualmente, solo se admite resource_name.</p>
not_tags	No	Array of Tags objects	<p>Especifica las etiquetas que no se incluyen. Cada etiqueta contiene un máximo de 10 claves, y cada clave contiene un máximo de 10 valores. El cuerpo de la estructura no puede faltar y la clave no se puede dejar en blanco o establecer una string vacía. Las claves deben ser únicas y los valores de una clave deben ser únicos. Los recursos no identificados por diferentes claves están en relación AND, y los valores en una etiqueta están en relación OR. Si no se especifica ninguna condición de filtrado, se devuelven los datos completos.</p>

Parámetro	Obligatorio	Tipo	Descripción
tags	No	Array of Tags objects	Especifica las etiquetas que se incluyen. Cada etiqueta contiene un máximo de 10 claves, y cada clave contiene un máximo de 10 valores. El cuerpo de la estructura no puede faltar y la clave no se puede dejar en blanco o establecer una string vacía. Las claves deben ser únicas y los valores de una clave deben ser únicos. Los recursos identificados por diferentes claves están en relación AND, y los valores en una etiqueta están en relación OR. Si no se especifica ningún criterio de filtrado de etiquetas, se devuelven los datos completos.
tags_any	No	Array of Tags objects	Especifica cualquier etiqueta que se incluya. Cada etiqueta contiene un máximo de 10 claves, y cada clave contiene un máximo de 10 valores. El cuerpo de la estructura no puede faltar y la clave no se puede dejar en blanco o establecer una string vacía. Las claves deben ser únicas y los valores de una clave deben ser únicos. Los recursos identificados por diferentes claves están en relación OR, y los valores en una etiqueta están en relación OR. Si no se especifica ninguna condición de filtrado, se devuelven los datos completos.

Parámetro	Obligatorio	Tipo	Descripción
not_tags_any	No	Array of Tags objects	Especifica cualquier etiqueta que no esté incluida. Cada etiqueta contiene un máximo de 10 claves, y cada clave contiene un máximo de 10 valores. El cuerpo de la estructura no puede faltar y la clave no se puede dejar en blanco o establecer una string vacía. Las claves deben ser únicas y los valores de una clave deben ser únicos. Los recursos no identificados por diferentes claves están en relación OR, y los valores en una etiqueta están en relación OR. Si no se especifica ninguna condición de filtrado, se devuelven los datos completos.
sys_tags	No	Array of Tags objects	Solo los usuarios con el permiso op_service pueden usar este campo para filtrar recursos. Solo una estructura de etiquetas está contenida cuando esta API es invocada por Tag Management Service (TMS). key es _sys_enterprise_project_id y value es la lista de ID del proyecto empresarial. Cuando TMS invoca esta API, una clave solo puede tener un valor. 0 indica el proyecto de empresa predeterminado. sys_tags y las condiciones de filtrado de etiquetas de tenant (without_any_tag_tags , tags_any , not_tags y not_tags_any) no se pueden usar al mismo tiempo.

Tabla 5-160 Match

Parámetro	Mandatory	Type	Descripción
key	Sí	String	Especifica la clave de etiqueta. El valor puede ser dc-directconnect , dc-vgw o dc-vif .

Parámetro	Mandatory	Type	Descripción
value	Sí	String	Especifica el valor de etiqueta. Cada valor puede contener un máximo de 255 caracteres de Unicode.

Tabla 5-161 Etiquetas

Parámetro	Obligatorio	Tipo	Descripción
key	Sí	String	Especifica la clave de etiqueta. Cada clave de etiqueta puede contener un máximo de 127 caracteres de Unicode. Debe especificarse key .
values	Sí	Array of strings	Especifica la lista de valores de etiqueta. Cada valor puede contener un máximo de 255 caracteres de Unicode.

Parámetros de respuesta

Código de estado: 200

Tabla 5-162 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
resources	Array of Resource objects	Especifica la lista de recursos.
total_count	Integer	Especifica el número total de registros.
request_id	String	Especifica el ID de la solicitud.

Tabla 5-163 Recurso

Parámetro	Tipo	Descripción
resource_detail	Object	Especifica los detalles del recurso.
resource_id	String	Especifica el ID del recurso.
resource_name	String	Especifica el nombre del recurso. Este parámetro es una string vacía de forma predeterminada si no hay un nombre de recurso.

Parámetro	Tipo	Descripción
tags	Array of Tag objects	Especifica la lista de etiquetas consultadas. Si no hay ninguna etiqueta coincidente, se devuelve una matriz vacía.
sys_tags	Array of Tag objects	Especifica la lista de etiquetas consultadas. Si no hay ninguna etiqueta coincidente, se devuelve una matriz vacía.

Tabla 5-164 Etiqueta

Parámetro	Tipo	Descripción
key	String	Especifica la clave de etiqueta. La clave puede contener un máximo de 36 caracteres de Unicode, incluidos letras, dígitos, guiones (-) y guiones bajos (_). Mínimo: 0 Máximo: 36
value	String	Especifica el valor de etiqueta. El valor puede contener un máximo de 43 caracteres de Unicode, incluidos letras, dígitos, guiones (-), guiones bajos (_), y puntos (.). Mínimo: 0 Máximo: 43

Ejemplo de solicitudes

- Consulta de recursos (configuración de **action** en **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" ]  
} ]  
}
```

- **Consulta del número total de recursos (action establecida para 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"  
  } ]  
}
```

Ejemplo de respuestas

Código de estado: 200

Aceptar

```
{  
  "resources" : [ ],  
  "total_count" : 0,  
  "request_id" : "9a4f4dfc4fb2fc101e65bba07d908535"  
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

Java

- **Consulta de recursos (configuración de action en 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());
    }
}
}
```

- **Consulta del número total de recursos (action establecida para 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("coun
```



```
t));
    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

- **Consulta de recursos (configuración de *action* en *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)
```

- **Consulta del número total de recursos (action establecida para 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

- **Consulta de recursos (configuración de `action` en `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)
  }
}
```

- **Consulta del número total de recursos (action establecida para 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-45a7-8dlb-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)  
  }  
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
200	Aceptar

Códigos de error

Consulte [Códigos de error](#).

5.5 Gestión de cuotas

5.5.1 Consulta de cuotas de recursos

Función

Esta API se utiliza para consultar el uso de recursos, por ejemplo, cuántas conexiones e interfaces virtuales se han creado.

Método de invocación

Para obtener más información, consulte [Invocación a las API](#).

URI

GET /v3/{project_id}/dcaas/quotas

Tabla 5-165 Parámetros de enrutamiento

Parámetro	Obligatorio	Tipo	Descripción
project_id	Sí	String	Especifica el ID del proyecto.

Tabla 5-166 Parámetros de consulta

Parámetro	Obligatorio	Tipo	Descripción
type	No	Array	Especifica las cuotas de recursos. <ul style="list-style-type: none"> ● physicalConnect: cuota y uso de la conexión ● virtualInterface: cuota y uso de la interfaz virtual ● globalDcGateway: Cuota y uso del gateway global de DC ● peerLinkPerGdgw: Cuota y uso de los enlaces de pares establecidos con un gateway global de DC

Parámetros de solicitud

No hay

Parámetros de respuesta

Código de estado: 201

Tabla 5-167 Parámetros del cuerpo de respuesta

Parámetro	Tipo	Descripción
quotas	quotas object	Proporciona detalles de cuota usada.

Tabla 5-168 quotas

Parámetro	Tipo	Descripción
resources	Array of Info objects	Detalles de uso de diferentes cuotas de recursos.

Tabla 5-169 Info

Parámetro	Tipo	Descripción
type	String	Especifica el tipo de cuota.
quota	Long	Especifica la cuota disponible. El valor -1 indica que no hay límite de cuota.
used	Long	Especifica las cuotas utilizadas.
unit	String	Especifica la unidad de medida del uso de recursos.

Ejemplo de solicitudes

Consulta de cuotas

```
GET https://{dc_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/dcaas/quotas
```

Ejemplo de respuestas

Código de estado: 201

Se consultan las cuotas.

```
{
  "quotas" : {
    "resources" : [ {
      "type" : "direct_connnect",
      "quota" : 1,
      "used" : 0,
      "unit" : "count"
    } ]
  }
}
```

Ejemplo de código del SDK

El ejemplo de código del SDK es el siguiente.

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)
    }
}
```

Otro más

Para obtener un ejemplo de código del SDK de más lenguajes de programación, consulta la ficha Ejemplo de código en [API Explorer](#). El ejemplo de código del SDK se puede generar automáticamente.

Códigos de estado

Código de estado	Descripción
201	Se consultan las cuotas.

Códigos de error

Consulte [Códigos de error](#).

6 Permisos y acciones admitidas

6.1 Introducción

Puede usar Identity and Access Management (IAM) para la gestión detallada de permisos de sus recursos de Direct Connect. Si su ID de HUAWEI no necesita usuarios individuales de IAM, puede omitir este tema.

Con IAM, puede controlar el acceso a recursos específicos en la nube. IAM admite la autorización basada en roles/políticas y la autorización basada en políticas de identidad.

En la siguiente tabla se describen las diferencias entre los dos modelos de autorización.

Tabla 6-1 Differences between role/policy-based and identity policy-based authorization

Name	Authorization Using	Permissions	Authorization Method	Scenario
Role/Policy	User-permission-authorization scope	<ul style="list-style-type: none"> ● System - defined roles ● System - defined policies ● Custom policies 	Assigning roles or policies to principals	To authorize a user, you need to add it to a user group first and then specify the scope of authorization. It is hard to provide fine-grained permissions control using authorization by user groups and a limited number of condition keys. This method is suitable for small- and medium-sized enterprises.

Name	Authorization Using	Permissions	Authorization Method	Scenario
Identity policy	Policies	<ul style="list-style-type: none"> ● System - defined identity policies ● Custom policies 	<ul style="list-style-type: none"> ● Assigning identity policies to principals ● Attaching identity policies to principals 	You can authorize a user by attaching an identity policy to it. User-specific authorization and a variety of key conditions allow for more fine-grained permissions control. However, this model can be hard to set up. It requires a certain amount of expertise and is suitable for medium- and large-sized enterprises.

Supongamos que desea conceder a los usuarios de IAM el permiso para crear los ECS en CN North-Beijing4 y buckets de OBS en CN South-Guangzhou. Con la autorización basada en roles/políticas, el administrador debe crear dos políticas personalizadas y adjuntar ambas a los usuarios de IAM. Con la autorización basada en políticas de identidad, el administrador solo necesita crear una política personalizada y configurar la clave de condición **g:RequestedRegion** para la política y, a continuación, adjuntar la política a los usuarios o conceder a los usuarios los permisos de acceso a las regiones especificadas. La autorización basada en políticas de identidad es más flexible que la autorización basada en roles/políticas.

Las políticas y acciones de los dos modelos de autorización no son interoperables. Se recomienda utilizar el modelo de autorización basado en políticas de identidad.

Si usa usuarios de IAM en su cuenta para invocar a una API, se deben conceder los permisos necesarios a los usuarios de IAM. Los permisos necesarios para invocar a una API están determinados por las acciones admitidas por la API. Solo los usuarios a los que se les han concedido permisos para permitir las acciones pueden invocar a la API con éxito.

Supongamos que un usuario de IAM desea invocar a una API para consultar conexiones. Con la autorización basada en roles/políticas, el usuario IAM debe disponer de los permisos que le permitan la acción **dcaas:directConnect:list**. Con la autorización basada en políticas de identidad, el usuario IAM debe disponer de los permisos que le permitan la acción **dcaas:directConnect:list**.

6.2 Acciones apoyadas por la autorización basada en roles/políticas

En este tema se describen las acciones admitidas por Direct Connect en la autorización basada en rol o política.

Acciones admitidas

IAM proporciona políticas definidas por el sistema que se pueden utilizar directamente. También puede crear políticas personalizadas para trabajar con políticas definidas por el sistema para un control de acceso más refinado. Las operaciones admitidas por las políticas son específicas de las API. Los siguientes son conceptos comunes relacionados con las políticas:

- Permisos: Instrucciones de una política que permiten o niegan ciertas operaciones.
- APIs: APIs REST que pueden ser llamadas por un usuario al que se le han concedido permisos específicos.
- Acciones: Operaciones específicas que están permitidas o denegadas.
- Acciones relacionadas: acciones de las que depende una acción específica para que surtan efecto. Al asignar permisos para la acción a un usuario, también debe asignar permisos para las acciones relacionadas.
- IAM o proyectos de empresa: Tipo de proyectos para los que una acción tendrá efecto. Las políticas que contienen acciones tanto para IAM como para proyectos empresariales pueden utilizarse y surtir efecto tanto para IAM como para Enterprise Management. Las políticas que solo contienen acciones para proyectos de IAM se pueden usar y solo tienen efecto para IAM. Los administradores pueden comprobar si una acción admite proyectos de IAM o proyectos de empresa en la lista de acciones. Para obtener más información sobre las diferencias entre IAM y la gestión empresarial, consulte [¿Cuáles son las diferencias entre IAM y la gestión empresarial?](#)

Direct Connect admite las siguientes acciones en políticas personalizadas:

- **Acciones permitidas para las conexiones:** contiene acciones admitidas por las API de conexiones, como las API para crear, consultar, eliminar y actualizar una conexión, y la API para consultar la lista de conexiones.
- **Acciones permitidas para gateway virtual:** contiene acciones admitidas por las API de los gateway virtuales, como las API para crear, consultar, eliminar y actualizar un gateway virtual, y la API para consultar la lista de gateway virtual.
- **Acciones permitidas para las interfaces virtuales:** contiene acciones admitidas por las API de interfaces virtuales, como las API para crear, consultar, eliminar y actualizar una interfaz virtual, y la API para consultar la lista de interfaces virtuales.
- **Acciones permitidas para las etiquetas:** contiene acciones admitidas por las API de etiquetas, como las API para agregar una etiqueta, consultar etiquetas de proyecto, consultar etiquetas de recursos, agregar etiquetas de recursos, eliminar etiquetas de recursos y consultar un recurso por etiqueta.

Acciones permitidas para las conexiones

Tabla 6-2 Acciones permitidas para conexiones

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Consulta de detalles de una conexión	GET /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}	dcaas:directConnect:get	-	√	×

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Actualización de una conexión	PUT /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}	dcaas:directConnect:update	-	√	×
Eliminación de una conexión	DELETE /v3/{project_id}/dcaas/direct-connects/{direct_connect_id}	dcaas:directConnect:delete	-	√	×
Consulta de la lista de conexiones	GET /v3/{project_id}/dcaas/direct-connects	dcaas:directConnect:list	-	√	×
Consulta de la lista de conexiones alojadas	GET /v3/{project_id}/dcaas/hosted-connects	dcaas:directConnect:listHostedDirectConnect	-	√	×
Creación de una conexión alojada	POST /v3/{project_id}/dcaas/hosted-connects	dcaas:directConnect:createHostedDirectConnect	-	√	×
Consulta de detalles de una conexión alojada	GET /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}	dcaas:directConnect:getHostedDirectConnect	-	√	×
Actualización de una conexión alojada	PUT /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}	dcaas:directConnect:updateHostedDirectConnect	-	√	×

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Eliminación de una conexión alojada	DELETE /v3/{project_id}/dcaas/hosted-connects/{hosted_connect_id}	dcaas:directConnect:deleteHostedDirectConnect	-	√	×

Acciones permitidas para gateway virtual

Tabla 6-3 Acciones permitidas para gateways virtuales

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Creación de un gateway virtual	POST /v3/{project_id}/dcaas/virtual-gateways	dcaas:vgw:create	vpc:vpcs:get vpc:vpcs:list vpc:subnets:get	√	×
Consulta de la lista del gateway virtual	GET /v3/{project_id}/dcaas/virtual-gateways	dcaas:vgw:list	-	√	×
Eliminación de un gateway virtual	DELETE /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}	dcaas:vgw:delete	-	√	×
Consulta de detalles de un gateway virtual	GET /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}	dcaas:vgw:get	-	√	×
Actualización de un gateway virtual	PUT /v3/{project_id}/dcaas/virtual-gateways/{virtual_gateway_id}	dcaas:vgw:update	-	√	×

Acciones permitidas para las interfaces virtuales

Tabla 6-4 Acciones permitidas para interfaces virtuales

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Creación de una interfaz virtual	POST /v3/{project_id}/dcaas/virtual-interfaces	dcaas:vif:create	vpc:vpcs:get vpc:vpcs:list vpc:subnets:get	√	×
Consulta de la lista de interfaz virtual	GET /v3/{project_id}/dcaas/virtual-interfaces	dcaas:vif:list	-	√	×
Eliminación de una interfaz virtual	DELETE /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}	dcaas:vif:delete	-	√	×
Consulta de detalles de una interfaz virtual	GET /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}	dcaas:vif:get	-	√	×
Actualización de una interfaz virtual	PUT /v3/{project_id}/dcaas/virtual-interfaces/{virtual_interface_id}	dcaas:vif:update	-	√	×

Acciones permitidas para las etiquetas

Tabla 6-5 Acciones permitidas para las etiquetas

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Consulta de etiquetas por tipo de recurso	GET /v3/{project_id}/{resource_type}/tags	dcaas:resources:listTag	-	√	×

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Consulta de etiquetas de recurso	GET /v3/{project_id}/{resource_type}/{resource_id}/tags	dcaas:resources:listResourceTag	-	√	×
Adición de una etiqueta de recurso	POST /v3/{project_id}/{resource_type}/{resource_id}/tags	dcaas:resources:tag	-	√	×
Adición o eliminación de etiquetas de recursos	POST /v3/{project_id}/{resource_type}/{resource_id}/tags/action	dcaas:resources:batchTagUntag	-	√	×
Eliminación de una etiqueta de recurso	DELETE /v3/{project_id}/{resource_type}/{resource_id}/tags/{key}	dcaas:resources:unTag	-	√	×
Consulta de recursos por etiqueta	POST /v3/{project_id}/{resource_type}/resource-instances/action	dcaas:resources:listByTag	-	√	×

Acciones permitidas para par de enlaces

Tabla 6-6 Acciones permitidas para par de enlaces

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Consultar la lista de pares de enlaces	GET /v3/{project_id}/dcaas/global-dc-gateways/{global_dc_gateway_id}/peer-links	dcaas:gdgw:listPeerlink	-	√	×

Permiso	API	Acción	Acción relacionada	Proyectos de IAM	Proyecto empresarial
Creación de un par de enlaces	POST /v3/{project_id}/dcaas/global-dc-gateways/{global_dc_gateway_id}/peer-links	dcaas:gdgw:createPeerlink	vpc:vpcs:get vpc:vpcs:list vpc:subnets:get	√	×
Consultar detalles de un par de enlaces	GET /v3/{project_id}/dcaas/global-dc-gateways/{global_dc_gateway_id}/peer-links/{peer_link_id}	dcaas:gdgw:getPeerlink	-	√	×
Actualización de un par de enlaces	PUT /v3/{project_id}/dcaas/global-dc-gateways/{global_dc_gateway_id}/peer-links/{peer_link_id}	dcaas:gdgw:updatePeerlink	-	√	×
Eliminación de un par de enlaces	DELETE /v3/{project_id}/dcaas/global-dc-gateways/{global_dc_gateway_id}/peer-links/{peer_link_id}	dcaas:gdgw:deletePeerlink	-	√	×

7 Parámetros públicos

7.1 Códigos de estado comunes

Respuesta exitosa	Mensaje	Descripción
200	OK	Código de respuesta normal para las operaciones GET y PUT
201	Created	Código de respuesta normal para operaciones POST
204	No Content	Código de respuesta normal para las operaciones DELETE

Respuesta de error	Descripción
400 Bad Request (400: Solicitud errónea)	Se produjo un error en el servidor al procesar la solicitud.
401 Unauthorized (401: No autorizado)	Debe introducir el nombre de usuario y la contraseña para acceder a la página solicitada.
403 Forbidden (403: Prohibido)	Está prohibido el acceso a la página solicitada.
404 Not Found (404: No se encontró la página)	No se pudo encontrar la página solicitada.
405 Method Not Allowed (405: Método no permitido)	No se le permite utilizar el método especificado en la solicitud.
406 Not Acceptable (406: No aceptable)	El cliente no pudo recibir la respuesta generada por el servidor.
407 Proxy Authentication Required (407: Autenticación de proxy requerida)	Debe utilizar el servidor proxy para la autenticación para que la solicitud pueda procesarse.

Respuesta de error	Descripción
408 Request Timeout (408: Se agotó el tiempo de espera de la solicitud)	Se agotó el tiempo de espera de la solicitud.
409 Conflict (409: Conflicto)	La solicitud no se pudo procesar debido a un conflicto.
500 Internal Server Error (500: Error interno del servidor)	La solicitud no se completa porque el servicio es anormal.
501 Not Implemented (501: No implementado)	La solicitud no se completa porque el servidor no admite la función solicitada.
502 Bad Gateway (502: Gateway incorrecto)	La solicitud no se completa porque el servidor recibe una respuesta no válida de un servidor ascendente.
503 Service Unavailable (503: Servicio no disponible)	La solicitud no se completa porque el sistema está fuera de servicio temporalmente.
504 Gateway Timeout (504: Tiempo de espera de gateway)	Se agotó el tiempo de espera de un gateway.

7.2 Códigos de errores

Código de estado	Código de error	Descripción	Solución
400	DC.0000	El cuerpo de la solicitud es anormal.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0001	Error de parámetros de solicitud.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0002	El recurso no existe.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0003	Error del sistema.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0004	Dirección IP no válida.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0005	Error al invocar a la API de VPC.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.

Código de estado	Código de error	Descripción	Solución
400	DC.0006	Error al invocar a la API de MO.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0007	La VPC no existe.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0008	El ASN de BGP está fuera de rango.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0009	La política no válida.	Contacte con el servicio de asistencia técnica.
400	DC.0010	El VPC_VNI no existe.	Contacte con el servicio de asistencia técnica.
400	DC.0011	El AGENT es anormal.	Corrija la configuración de los parámetros y envíe la solicitud de nuevo.
400	DC.0012	Error al invocar a la API de CBC.	Contacte con el servicio de asistencia técnica.
400	DC.0013	No tiene permiso para operar el campo.	Contacte con el servicio de asistencia técnica.
400	DC.1000	Ancho de banda de conexión insuficiente.	Contacte con el servicio de asistencia técnica.
400	DC.1001	Se especifica una VLAN para una conexión no alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1002	Error al crear una conexión alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1003	No se especifica una VLAN para una conexión alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1004	Se selecciona una conexión sin operaciones para crear una conexión alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1005	La conexión es anormal.	Contacte con el servicio de asistencia técnica.
400	DC.1006	El dispositivo y el tipo no se pueden modificar cuando la conexión está en uso.	Contacte con el servicio de asistencia técnica.
400	DC.1007	La conexión está ocupada.	Contacte con el servicio de asistencia técnica.

Código de estado	Código de error	Descripción	Solución
400	DC.1008	Las VLAN de conexión alojada entran en conflicto.	Contacte con el servicio de asistencia técnica.
400	DC.1010	Se reduce el ancho de banda de la conexión.	Contacte con el servicio de asistencia técnica.
400	DC.1011	No se puede cambiar el ancho de banda de la conexión.	Contacte con el servicio de asistencia técnica.
400	DC.1012	La conexión no existe.	Contacte con el servicio de asistencia técnica.
400	DC.1014	Se establece una conexión redundante para una conexión no estándar.	Contacte con el servicio de asistencia técnica.
400	DC.1015	La conexión o LAG no existe.	Contacte con el servicio de asistencia técnica.
400	DC.1017	El ID de VLAN de la conexión alojada es 0.	Contacte con el servicio de asistencia técnica.
400	DC.1019	La cuota de conexión se ha agotado.	Contacte con el servicio de asistencia técnica.
400	DC.1100	El ID de grupo de punto de conexión local del gateway virtual no es válido.	Contacte con el servicio de asistencia técnica.
400	DC.1101	El número de subredes locales excede el límite.	Contacte con el servicio de asistencia técnica.
400	DC.1102	El dispositivo en standby se especifica cuando no se especifica el dispositivo activo.	Contacte con el servicio de asistencia técnica.
400	DC.1103	Los dispositivos activos y en standby son los mismos.	Contacte con el servicio de asistencia técnica.
400	DC.1104	La información del dispositivo no se puede actualizar para el gateway virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1105	El bloque CIDR local se superpone con el bloque CIDR remoto.	Contacte con el servicio de asistencia técnica.
400	DC.1106	El gateway virtual está ocupado.	Contacte con el servicio de asistencia técnica.
400	DC.1107	Las VLAN del gateway virtual están agotadas.	Contacte con el servicio de asistencia técnica.

Código de estado	Código de error	Descripción	Solución
400	DC.1108	Los VNI del gateway virtual están agotados.	Contacte con el servicio de asistencia técnica.
400	DC.1109	Los VRF del gateway virtual están agotados.	Contacte con el servicio de asistencia técnica.
400	DC.1110	Se ha asociado un gateway virtual con la VPC.	Contacte con el servicio de asistencia técnica.
400	DC.1111	La gateway virtual no existe.	Contacte con el servicio de asistencia técnica.
400	DC.1112	Los dispositivos utilizados por el gateway virtual no están en modo activo/en espera.	Contacte con el servicio de asistencia técnica.
400	DC.1113	El gateway virtual está en el estado de corte.	Contacte con el servicio de asistencia técnica.
400	DC.1114	traffic_mode no se puede actualizar para VLAN o gateways virtuales con un solo dispositivo VTEP.	Contacte con el servicio de asistencia técnica.
400	DC.1115	El VNI del gateway virtual no existe.	Contacte con el servicio de asistencia técnica.
400	DC.1116	Error al actualizar el gateway de enrutamiento.	Contacte con el servicio de asistencia técnica.
400	DC.1117	El gateway virtual se ha asociado con dos conexiones y no se puede asociar con más conexiones.	Contacte con el servicio de asistencia técnica.
400	DC.1118	El gateway virtual está siendo operado (bloqueado).	Contacte con el servicio de asistencia técnica.
400	DC.1119	Error al crear el grupo de dispositivos de gateway de enrutamiento.	Contacte con el servicio de asistencia técnica.
400	DC.1120	El elemento de configuración main_az_list es incorrecto.	Contacte con el servicio de asistencia técnica.
400	DC.1200	No se especifica ningún LAG o conexión para la interfaz virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1201	La conexión no coincide con el tipo del gateway virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1202	No se pueden crear las interfaces virtuales del tipo double ipsec .	Contacte con el servicio de asistencia técnica.

Código de estado	Código de error	Descripción	Solución
400	DC.1203	No se especifica ningún ASN para la interfaz virtual de BGP.	Contacte con el servicio de asistencia técnica.
400	DC.1204	La creación de interfaces virtuales no es compatible con las conexiones de operaciones.	Contacte con el servicio de asistencia técnica.
400	DC.1205	El estado del recurso asociado con la interfaz virtual es anormal.	Contacte con el servicio de asistencia técnica.
400	DC.1206	La conexión alojada se ha asociado con una interfaz virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1207	La VLAN de la interfaz virtual es incompatible con la de la conexión alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1208	No se puede modificar el ancho de banda de la interfaz virtual asociada a una conexión alojada.	Contacte con el servicio de asistencia técnica.
400	DC.1209	Conflicto de VLAN de interfaz virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1210	No se puede cambiar el estado de la interfaz virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1211	La interfaz virtual no existe.	Contacte con el servicio de asistencia técnica.
400	DC.1212	El gateway virtual está en un grupo diferente de la conexión.	Contacte con el servicio de asistencia técnica.
400	DC.1213	La interfaz virtual no coincide con el tenant.	Contacte con el servicio de asistencia técnica.
400	DC.1214	Se ha utilizado la interfaz virtual.	Contacte con el servicio de asistencia técnica.
400	DC.1215	El tipo de la interfaz virtual no coincide.	Contacte con el servicio de asistencia técnica.
400	DC.1216	El gateway local no es válido.	Contacte con el servicio de asistencia técnica.
400	DC.1217	El ID de sitio de borde de IES de la conexión no coincide con el del gateway local.	Contacte con el servicio de asistencia técnica.
400	DC.1218	El gateway local no coincide.	Contacte con el servicio de asistencia técnica.

Código de estado	Código de error	Descripción	Solución
400	DC.1219	Se crea una interfaz virtual junto con VLAN 0.	Contacte con el servicio de asistencia técnica.
400	DC.1220	El campo vif email de la red pública está vacío.	Contacte con el servicio de asistencia técnica.
400	DC.1221	El dispositivo del gateway virtual seleccionado no admite el acceso desde las ubicaciones de Direct Connect.	Contacte con el servicio de asistencia técnica.
400	DC.1222	Insuficientes VNI POP.	Contacte con el servicio de asistencia técnica.
400	DC.1223	El ASN de BGP local es el mismo que el ASN de BGP remoto.	Contacte con el servicio de asistencia técnica.
400	DC.1224	La interfaz virtual de la conexión terminada en una ubicación de Direct Connect no admite VLAN 0.	Contacte con el servicio de asistencia técnica.
400	DC.1225	El número de interfaces virtuales para las que está habilitada la recopilación de estadísticas de tráfico alcanza el límite.	Contacte con el servicio de asistencia técnica.
400	DC.1226	La recopilación de estadísticas de tráfico no se puede habilitar en las interfaces remotas de Capa 2 que no tienen subinterfaces.	Contacte con el servicio de asistencia técnica.
400	DC.1227	Las estadísticas de tráfico no se pueden actualizar porque la interfaz virtual es anormal.	Contacte con el servicio de asistencia técnica.
400	DC.1400	El número de grupos de punto de conexión alcanza el límite.	Contacte con el servicio de asistencia técnica.
400	DC.1401	Bloques CIDR duplicados.	Contacte con el servicio de asistencia técnica.
400	DC.1402	El grupo de puntos de conexión está en uso.	Contacte con el servicio de asistencia técnica.
400	DC.1403	El grupo de punto de conexión no existe.	Contacte con el servicio de asistencia técnica.

A Apéndices

A.1 Obtención de un ID de proyecto

Escenarios

Se requiere un ID de proyecto para algunas URL cuando se llama a una API. Por lo tanto, es necesario obtener un ID de proyecto por adelantado. Hay dos métodos disponibles:

- [Obtener el ID del proyecto llamando a una API](#)
- [Obtener el ID del proyecto desde la consola](#)

Obtener el ID del proyecto llamando a una API

Puede obtener el ID del proyecto llamando a la API utilizada para [consultar proyectos en función de criterios especificados](#).

La API utilizada para obtener un ID de proyecto es GET `https://{Endpoint}/v3/projects`. `{Endpoint}` es el punto final de IAM y se puede obtener del . Para obtener más información sobre la autenticación de API, consulte [Autenticación](#).

El siguiente es un ejemplo de respuesta. El valor de `id` es el ID del proyecto.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/
a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
  }
}
```

```
"previous": null,  
"self": "https://www.example.com/v3/projects"  
}
```

Obtener un ID de proyecto desde la consola

Para obtener un ID de proyecto de la consola, realice las siguientes operaciones:

1. Inicie sesión en la consola de gestión.
2. Haga clic en el nombre de usuario y seleccione **My Credentials** en la lista desplegable.
En la página **API Credentials**, vea el ID de proyecto en la lista de proyectos.

Figura A-1 Consulta del ID del proyecto

