

Elastic IP

API Reference (ME-Abu Dhabi Region)

Issue 01
Date 2020-11-05



Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Before You Start.....	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 EIP Endpoints.....	1
1.4 Notes and Constraints.....	1
1.5 Concepts.....	2
2 API Overview.....	3
3 Calling APIs.....	5
3.1 Making an API Request.....	5
3.2 Authentication.....	9
3.3 Response.....	11
4 APIs.....	13
4.1 Elastic IP.....	13
4.1.1 Assigning an EIP.....	13
4.1.2 Querying an EIP.....	19
4.1.3 Querying EIPs.....	22
4.1.4 Updating an EIP.....	26
4.1.5 Releasing an EIP.....	30
4.2 Bandwidth.....	31
4.2.1 Querying a Bandwidth.....	31
4.2.2 Querying Bandwidths.....	33
4.2.3 Updating a Bandwidth.....	37
4.3 Bandwidth (V2.0).....	41
4.3.1 Assigning a Shared Bandwidth.....	41
4.3.2 Deleting a Shared Bandwidth.....	46
4.3.3 Adding an EIP to a Shared Bandwidth.....	47
4.3.4 Removing an EIP from a Shared Bandwidth.....	51
4.4 Quota.....	54
4.4.1 Querying the Quota.....	54
4.5 EIP Tag Management.....	58
4.5.1 Creating a Tag for an EIP.....	58
4.5.2 Querying EIP Tags.....	59

4.5.3 Deleting an EIP Tag.....	61
4.5.4 Batch Creating or Deleting EIP Tags.....	62
4.5.5 Querying EIPs by Tag.....	65
4.5.6 Querying EIP Tags in a Specified Project.....	70
5 Native OpenStack Neutron APIs V2.0.....	72
5.1 API Version Information.....	72
5.1.1 Querying API Versions.....	72
5.1.2 Pagination.....	74
5.2 Floating IP Address.....	76
5.2.1 Querying Floating IP Addresses.....	76
5.2.2 Querying a Floating IP Address.....	80
5.2.3 Assigning a Floating IP Address.....	82
5.2.4 Updating a Floating IP Address.....	85
5.2.5 Deleting a Floating IP Address.....	88
6 Permissions Policies and Supported Actions.....	90
6.1 Elastic IP.....	90
6.2 Bandwidth.....	90
6.3 Bandwidth (V2.0).....	91
6.4 EIP Tags.....	91
6.5 Floating IP Address (OpenStack Neutron API).....	92
6.6 Precautions for API Permissions.....	92
7 Public Parameters.....	93
7.1 Status Codes.....	93
7.2 Error Codes.....	94
A Appendix.....	102
A.1 VPC Monitoring Metrics.....	102
A.2 Obtaining a Project ID.....	104
B Change History.....	106

1 Before You Start

1.1 Overview

Welcome to *Elastic IP API Reference*. The EIP service provides independent public IP addresses and bandwidth for Internet access. EIPs can be bound to or unbound from ECSs, virtual IP addresses, load balancers, and NAT gateways. Various billing modes are provided to meet diverse service requirements.

This document describes how to use application programming interfaces (APIs) to perform operations on EIPs, such as creating, querying, deleting, and updating an EIP. For details about all supported operations, see [2 API Overview](#).

If you plan to access EIPs through an API, ensure that you are familiar with EIP concepts. For details, see "Service Overview" in the *Elastic IP User Guide*.

1.2 API Calling

EIP supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [3 Calling APIs](#).

1.3 EIP Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. Currently, EIP and VPC use same endpoints. For the endpoints of all services, see [Regions and Endpoints](#).

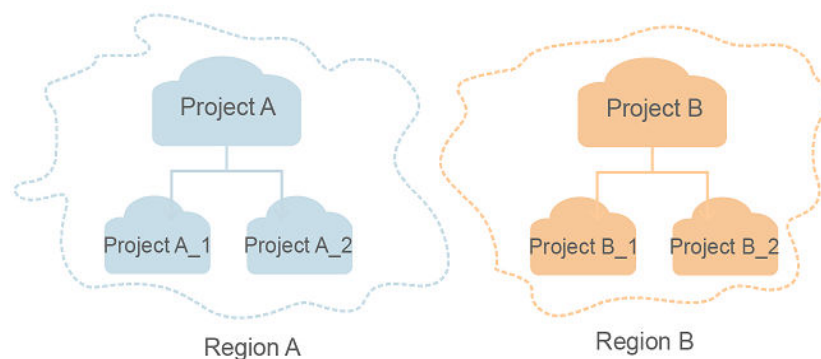
1.4 Notes and Constraints

The number of EIPs that you can assign is determined by your quota. To view or increase the quota, see "What Is a Quota?" in *Elastic IP User Guide*.

1.5 Concepts

- **Account**
An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. To ensure account security, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- **User**
An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).
API authentication requires information such as the account name, username, and password.
- **Region**
A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.
- **AZ**
An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- **Project**
A project corresponds to a region. Default projects are defined to a group and have physically isolated resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources in the region under their accounts. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



2 API Overview

APIs provided by the EIP service include native OpenStack APIs and EIP APIs.

A combination of these two types of APIs allows you to use all functions provided by the EIP service.

Table 2-1 EIP APIs

Type	Subtype	Description
EIP API	EIP	APIs for assigning, querying, updating, and releasing EIPs
EIP API	Bandwidth	APIs for querying and updating bandwidth
EIP API	Bandwidth (v2.0)	<ul style="list-style-type: none">• APIs for assigning and deleting shared bandwidth• APIs for adding an EIP to or removing an EIP from a shared bandwidth
EIP API	Quota	API for querying quota values
EIP API	EIP tag management	APIs for adding tags to EIPs as well as querying and deleting EIP tags
OpenStack Neutron API	Floating IP address	APIs for assigning, querying, updating, and releasing floating IP addresses

Type	Subtype	Description
OpenStack Neutron API	API version	APIs for querying all available API versions and displaying the results in pages.

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

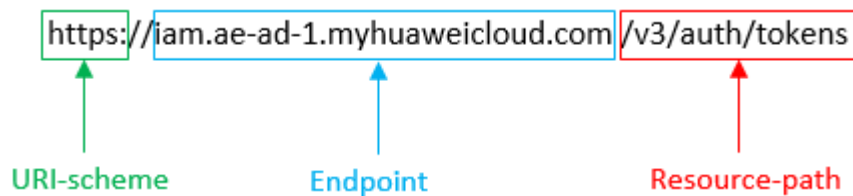
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM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

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

For example, to obtain an IAM token in the **UAE-Abu Dhabi** region, obtain the endpoint of IAM (**iam.ae-ad-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

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

For example, in the case of the API used to [obtain a user token](#), the request method is **POST**. The request is as follows:

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
```

Request Header

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

Common request header fields are as follows.

Table 3-3 Common request header fields

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

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in A.2 Obtaining a Project ID .	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (only this API does not require authentication). After the request is processed, the value of X-Subject-Token in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZlhvcNAQcCo...ggg1BBIINPXsidG9rZ

 **NOTE**

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

For more details, see "Authentication Using AK/SK" in [3.2 Authentication](#).

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

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

(Optional) Request Body

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

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

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

 NOTE

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

```
POST https://iam.ae-ad-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": "xxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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

3.2 Authentication

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure than token-based authentication.

Token-based Authentication

 NOTE

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

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

You can obtain a token by calling the [Obtaining User Token](#) API. When you call the API, set **auth.scope** in the request body to **project**.

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

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

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

AK/SK-based Authentication

NOTE

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

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

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

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

NOTE

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

3.3 Response

Status Code

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

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [7.1 Status Codes](#).

For example, if status code **201** is returned for calling the API used to [obtain a user token](#), the request is successful.

Response Header

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

[Figure 3-2](#) shows the response header fields for the API used to [obtain a user token](#). The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

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

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token
→ MIIVXQYJKoZIhvcNAQcCoIIYJCCEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6ijlwMTktMDItMTNUMC
fj3KIs6YgKnpVNRbW2eZ5eb78SZOkqjACgkIQ1wi4JlGzrpd18LGXK5tdf4q4qHCYb8P4NaY0NYejcAgzJVeFYtLWT1GSO0zxKZmlQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRC9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqglFkNPQuFSOUB+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbvpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

(Optional) Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to [obtain a user token](#).

```
{
  "token": {
```

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

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

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

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

4 APIs

4.1 Elastic IP

4.1.1 Assigning an EIP

Function

This API is used to assign an EIP.

The EIP service provides independent public IP addresses and bandwidth for Internet access. EIPs can be bound to or unbound from ECSs, BMSs, virtual IP addresses, load balancers, and NAT gateways.

URI

POST /v1/{project_id}/publicips

[Table 4-1](#) describes the parameters.

Table 4-1 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .

Request Message

- Request parameter

Table 4-2 Request parameter

Name	Mandatory	Type	Description
publicip	Yes	Object	Specifies the EIP object. For details, see Table 4-3 .
bandwidth	Yes	Object	Specifies the bandwidth object. For details, see Table 4-4 .

Table 4-3 Description of the **publicip** field

Name	Mandatory	Type	Description
type	Yes	String	<ul style="list-style-type: none"> Specifies the EIP type. The value can be 5_bgp. Constraints: <ul style="list-style-type: none"> The configured value must be supported by the system. publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.
ip_address	No	String	<ul style="list-style-type: none"> Specifies the EIP to be assigned. The system automatically assigns an EIP if you do not specify it. The value must be a valid IPv4 address in the available IP address range.

Table 4-4 Description of the **bandwidth** field

Name	Mandatory	Type	Description
name	Yes	String	<ul style="list-style-type: none">• Specifies the bandwidth name.• The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).• This parameter is mandatory when share_type is set to PER. This parameter will be ignored when share_type is set to WHOLE with an ID specified.

Name	Mandatory	Type	Description
size	Yes	Integer	<ul style="list-style-type: none"> • Specifies the bandwidth size. • The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.) • This parameter is mandatory when share_type is set to PER. This parameter will be ignored when share_type is set to WHOLE with an ID specified. • The minimum increment for bandwidth adjustment varies depending on the bandwidth range. The details are as follows: <ul style="list-style-type: none"> - The minimum increment is 1 Mbit/s if the allowed bandwidth ranges from 0 Mbit/s to 300 Mbit/s (with 300 Mbit/s included). - The minimum increment is 50 Mbit/s if the allowed bandwidth

Name	Mandator y	Type	Description
			<p>ranges from 300 Mbit/s to 1000 Mbit/s (with 1000 Mbit/s included).</p> <ul style="list-style-type: none"> - The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1000 Mbit/s.
share_type	Yes	String	<ul style="list-style-type: none"> • Specifies the bandwidth type. • The value is PER, indicating that the bandwidth is dedicated.
charge_mode	No	String	<ul style="list-style-type: none"> • The value is traffic, indicating that the billing is based on traffic.

- Example request 1 (IPv4 EIP with dedicated bandwidth)

POST `https://{Endpoint}/v1/{project_id}/publicips`

```
{
  "publicip": {
    "type": "5_bgp",
    "ip_version": 4
  },
  "bandwidth": {
    "name": "bandwidth123",
    "size": 10,
    "share_type": "PER"
  }
}
```

Response Message

- Response parameter

Table 4-5 Response parameter

Name	Type	Description
publicip	Object	Specifies the EIP object. For details, see Table 4-6 .

Table 4-6 Description of the **publicip** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.
status	String	<ul style="list-style-type: none"> • Specifies the EIP status. • Possible values are as follows: <ul style="list-style-type: none"> - FREEZED (Frozen) - BIND_ERROR (Binding failed) - BINDING (Binding) - PENDING_DELETE (Releasing) - PENDING_CREATE (Assigning) - PENDING_UPDATE (Updating) - DOWN (Unbound) - ACTIVE (Bound) - ELB (Bound to a load balancer) - ERROR (Exceptions)
type	String	<ul style="list-style-type: none"> • Specifies the EIP type. • The value can be 5_bgp. • Constraints: <ul style="list-style-type: none"> - The configured value must be supported by the system. - publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).

- Example response 1 (IPv4 EIP with dedicated bandwidth)

```
{
  "publicip": {
    "id": "f588ccfa-8750-4d7c-bf5d-2ede24414706",
    "status": "PENDING_CREATE",
    "type": "5_bgp",
    "public_ip_address": "161.xx.xx.7",
    "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
    "ip_version": 4,
    "create_time": "2015-07-16 04:10:52",
    "bandwidth_size": 0
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.1.2 Querying an EIP

Function

This API is used to query a specific EIP.

URI

GET /v1/{project_id}/publicips/{publicip_id}

[Table 4-7](#) describes the parameters.

Table 4-7 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter
None
- Example request
Get `https://{Endpoint}/v1/{project_id}/publicips/{publicip_id}`

Response Message

- Response parameter

Table 4-8 Response parameter

Name	Type	Description
publicip	Object	Specifies the EIP object. For details, see Table 4-9 .

Table 4-9 Description of the **publicip** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.
status	String	<ul style="list-style-type: none">• Specifies the EIP status.• Possible values are as follows:<ul style="list-style-type: none">- FREEZED (Frozen)- BIND_ERROR (Binding failed)- BINDING (Binding)- PENDING_DELETE (Releasing)- PENDING_CREATE (Assigning)- PENDING_UPDATE (Updating)- DOWN (Unbound)- ACTIVE (Bound)- ELB (Bound to a load balancer)- ERROR (Exceptions)

Name	Type	Description
type	String	<ul style="list-style-type: none"> • Specifies the EIP type. • The value can be 5_bgp. • Constraints: <ul style="list-style-type: none"> – The configured value must be supported by the system. – publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.
private_ip_address	String	<ul style="list-style-type: none"> • Specifies the private IP address bound to the EIP. • This parameter is returned only when a private IP address is bound to the EIP.
port_id	String	<ul style="list-style-type: none"> • Specifies the port ID. • This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned.
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
bandwidth_share_type	String	<ul style="list-style-type: none"> • Specifies the EIP bandwidth type. • The value can be PER or WHOLE. <ul style="list-style-type: none"> – PER: Dedicated bandwidth – WHOLE: Shared bandwidth

- Example response

```
{
  "publicip": {
    "id": "2ec9b78d-9368-46f3-8f29-d1a95622a568",
    "status": "DOWN",
    "type": "5_bgp",
    "public_ip_address": "161.xx.xx.12",
    "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
    "private_ip_address": "192.168.10.5",
    "create_time": "2015-07-16 04:32:50",
    "bandwidth_id": "49c8825b-bed9-46ff-9416-704b96d876a2",
    "bandwidth_share_type": "PER",
    "bandwidth_size": 10, //The EIP bandwidth size is 10 Mbit/s.

    "ip_version": 4
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.1.3 Querying EIPs

Function

This API is used to query EIPs.

URI

GET /v1/{project_id}/publicips

[Table 4-10](#) describes the parameters.

Table 4-10 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
marker	No	String	Specifies the start resource ID of pagination query. If the parameter is left blank, only resources on the first page are queried.

Name	Mandatory	Type	Description
limit	No	Integer	<ul style="list-style-type: none">Specifies the number of records returned on each page.The value ranges from 0 to intmax.

Request Message

- Request parameter
None
- Example request
GET https://{Endpoint}/v1/{project_id}/publicips?limit={limit}&marker={marker}

Response Message

- Response parameter

Table 4-11 Response parameter

Name	Type	Description
publicips	Array of publicips objects	Specifies the EIP object. For details, see Table 4-12 .

Table 4-12 Description of the **publicips** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.

Name	Type	Description
status	String	<ul style="list-style-type: none"> • Specifies the EIP status. • Possible values are as follows: <ul style="list-style-type: none"> - FREEZED (Frozen) - BIND_ERROR (Binding failed) - BINDING (Binding) - PENDING_DELETE (Releasing) - PENDING_CREATE (Assigning) - PENDING_UPDATE (Updating) - DOWN (Unbound) - ACTIVE (Bound) - ELB (Bound to a load balancer) - ERROR (Exceptions)
type	String	<ul style="list-style-type: none"> • Specifies the EIP type. • The value can be 5_bgp. • Constraints: <ul style="list-style-type: none"> - The configured value must be supported by the system. - publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.

Name	Type	Description
private_ip_address	String	<ul style="list-style-type: none"> Specifies the private IP address bound to the EIP. This parameter is returned only if the private IP address is bound to the EIP.
port_id	String	<ul style="list-style-type: none"> Specifies the port ID. This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned.
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
bandwidth_share_type	String	<ul style="list-style-type: none"> Specifies the EIP bandwidth type. The value can be PER or WHOLE. <ul style="list-style-type: none"> PER: Dedicated bandwidth WHOLE: Shared bandwidth

- Example response

```

{
  "publicips": [
    {
      "id": "6285e7be-fd9f-497c-bc2d-dd0bdea6efe0",
      "status": "DOWN",
      "type": "5_bgp",
      "public_ip_address": "161.xx.xx.9",
      "private_ip_address": "192.168.10.5",
      "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
      "create_time": "2015-07-16 04:22:32",
      "bandwidth_id": "3fa5b383-5a73-4dcb-a314-c6128546d855",
      "bandwidth_share_type": "PER",
      "bandwidth_size": 5,

      "ip_version": 4
    },
    {
      "id": "80d5b82e-43b9-4f82-809a-37bec5793bd4",
      "status": "DOWN",
      "type": "5_bgp",

```

```
"public_ip_address": "161.xx.xx.10",  
"private_ip_address": "192.168.10.6",  
"tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",  
"create_time": "2015-07-16 04:23:03",  
"bandwidth_id": "a79fd11a-047b-4f5b-8f12-99c178cc780a",  
"bandwidth_share_type": "PER",  
"bandwidth_size": 5  
  
"ip_version": 4  
}  
]  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.1.4 Updating an EIP

Function

This API is used to update an EIP, for example, binding an EIP to a NIC, unbinding an EIP from a NIC, or converting the IP address version.

URI

PUT /v1/{project_id}/publicips/{publicip_id}

[Table 4-13](#) describes the parameters.

Table 4-13 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

Table 4-14 Request parameter

Name	Mandatory	Type	Description
publicip	Yes	Object	Specifies the EIP object. For details, see Table 4-15 .

Table 4-15 Description of the **publicip** field

Name	Mandatory	Type	Description
port_id	No	String	<ul style="list-style-type: none"> Specifies the port ID. The value must be an existing port ID. If this parameter is not included or the parameter value is left blank, the EIP is unbound. If the specified port ID does not exist or has already been bound with an EIP, an error message will be displayed.

- Example request 1 (Binding an EIP to a NIC)
PUT https://{Endpoint}/v1/{project_id}/publicips/{publicip_id}

```

{
  "publicip": {
    "port_id": "f588ccfa-8750-4d7c-bf5d-2ede24414706"
  }
}

```

Response Message

- Response parameter

Table 4-16 Response parameter

Name	Type	Description
publicip	<i>Object</i>	Specifies the EIP object. For details, see Table 4-17 .

Table 4-17 Description of the **publicips** field

Name	Type	Description
id	String	Specifies the unique identifier of an EIP.
status	String	<ul style="list-style-type: none"> • Specifies the EIP status. • Possible values are as follows: <ul style="list-style-type: none"> - FREEZED (Frozen) - BIND_ERROR (Binding failed) - BINDING (Binding) - PENDING_DELETE (Releasing) - PENDING_CREATE (Assigning) - PENDING_UPDATE (Updating) - DOWN (Unbound) - ACTIVE (Bound) - ELB (Bound to a load balancer) - ERROR (Exceptions)
type	String	<ul style="list-style-type: none"> • Specifies the EIP type. • The value can be 5_bgp. • Constraints: <ul style="list-style-type: none"> - The configured value must be supported by the system. - publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.
public_ip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.

Name	Type	Description
private_ip_address	String	<ul style="list-style-type: none"> Specifies the private IP address bound to the EIP. This parameter is returned only when a port is associated with the EIP.
port_id	String	<ul style="list-style-type: none"> Specifies the port ID. This parameter is returned only when a port is associated with the EIP.
tenant_id	String	Specifies the project ID.
create_time	String	Specifies the time (UTC) when the EIP is assigned.
bandwidth_id	String	Specifies the ID of the EIP bandwidth.
bandwidth_size	Integer	Specifies the bandwidth (Mbit/s).
bandwidth_share_type	String	<ul style="list-style-type: none"> Specifies the EIP bandwidth type. The value can be PER or WHOLE. <ul style="list-style-type: none"> PER: Dedicated bandwidth WHOLE: Shared bandwidth

- Example response 1 (Binding an EIP to a NIC)

```
{
  "publicip": {
    "id": "f6318bef-6508-4ea5-a48f-6152b6b1a8fb",
    "status": "ACTIVE",
    "type": "5_bgp",
    "port_id": "a135e9b8-1630-40d2-a6c5-eb534a61efbe",
    "public_ip_address": "10.xx.xx.162",
    "private_ip_address": "192.168.1.131",
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "create_time": "2019-03-27 01:33:18",
    "bandwidth_size": 7,
    "ip_version": 4
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.1.5 Releasing an EIP

Function

This API is used to release an EIP.

URI

DELETE /v1/{project_id}/publicips/{publicip_id}

[Table 4-18](#) describes the parameters.

Table 4-18 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter
None
- Example request
DELETE https://{Endpoint}/v1/{project_id}/publicips

Response Message

- Response parameter
None
- Example response
None
Or

```
{  
  "code": "xxx",  
  "message": "xxxxx"  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.2 Bandwidth

4.2.1 Querying a Bandwidth

Function

This API is used to query details about a bandwidth.

URI

GET /v1/{project_id}/bandwidths/{bandwidth_id}

[Table 4-19](#) describes the parameters.

Table 4-19 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter
None
- Example request
Get `https://{Endpoint}/v1/{project_id}/bandwidths/{bandwidth_id}`

Response Message

- Response parameter

Table 4-20 Response parameter

Name	Type	Description
bandwidth	Object	Specifies the bandwidth object.

Table 4-21 Description of the **bandwidth** field

Name	Type	Description
name	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).
size	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size. The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the bandwidth range of each region on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	<ul style="list-style-type: none"> The value is PER, indicating that the bandwidth is dedicated.
publicip_info	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies information about the EIP that uses the bandwidth. For details, see Table 4-22.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	<ul style="list-style-type: none"> Specifies the bandwidth type. The value is bgp.
charge_mode	String	<ul style="list-style-type: none"> Specifies whether the billing is based on traffic or bandwidth. Possible values can be bandwidth (billed by bandwidth) and traffic (billed by traffic). If the value is an empty character string or no value is specified, value bandwidth is used.
status	String	<ul style="list-style-type: none"> Specifies the bandwidth status. Possible values are as follows: <ul style="list-style-type: none"> FREEZED (Frozen) NORMAL (Normal)

Table 4-22 publicip_info object

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	<ul style="list-style-type: none">Specifies the EIP type.The value can be 5_bgp.The configured value must be supported by the system.

- Example response

```
{
  "bandwidth": {
    "id": "3cbd5ae9-368f-4bc8-8841-f2ecc322c64a",
    "name": "EIPResourceSetup_1553594229",
    "size": 5,
    "share_type": "PER",
    "publicip_info": [
      {
        "publicip_id": "22b02f40-b95f-465a-ae9b-7c8b0f042a41",
        "publicip_address": "10.xx.xx.62",
        "ip_version": 4,
        "publicip_type": "5_bgp"
      }
    ],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "bgp",
    "charge_mode": "bandwidth",
    "status": "NORMAL"
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.2.2 Querying Bandwidths

Function

This API is used to query bandwidths using search criteria.

URI

GET /v1/{project_id}/bandwidths

[Table 4-23](#) describes the parameters.

Table 4-23 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
marker	No	String	Specifies the start resource ID of pagination query. If the parameter is left blank, only resources on the first page are queried.
limit	No	Integer	<ul style="list-style-type: none"> Specifies the number of records returned on each page. The value ranges from 0 to intmax.

Request Message

- Request parameter
None
- Example request
GET https://{Endpoint}/v1/{project_id}/bandwidths?limit={limit}&marker={marker}

Response Message

- Response parameter

Table 4-24 Response parameter

Name	Type	Description
bandwidths	Array of bandwidths objects	Specifies the bandwidth objects. For details, see Table 4-25 .

Table 4-25 Description of the **bandwidths** field

Name	Type	Description
name	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).

Name	Type	Description
size	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size in Mbit/s. The value ranges from 1 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	<ul style="list-style-type: none"> The value is PER, indicating that the bandwidth is dedicated. <p>If this parameter is not set, the list of all bandwidths will be returned by default.</p>
publicip_info	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies the bandwidth information. For details, see Table 4-26.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	<ul style="list-style-type: none"> Specifies the bandwidth type. The value is bgp.
charge_mode	String	<ul style="list-style-type: none"> Specifies whether the bandwidth is billed by traffic or by bandwidth size. Possible values can be bandwidth (billed by bandwidth) and traffic (billed by traffic). If the value is an empty character string or no value is specified, value bandwidth is used.
status	String	<ul style="list-style-type: none"> Specifies the bandwidth status. Possible values are as follows: <ul style="list-style-type: none"> FREEZED (Frozen) NORMAL (Normal)

Table 4-26 publicip_info object

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	<ul style="list-style-type: none"> Specifies the EIP type. The value can be 5_bgp. The configured value must be supported by the system.

- Example response

```
{
  "bandwidths": [
    {
      "id": "09b99c91-da7c-449f-94e2-f4934c5b2a71",
      "name": "vpngw-f632a7b0-ef50-4ac5-97e9-ddc56b3d5977",
      "size": 2000,
      "share_type": "PER",
      "publicip_info": [
        {
          "publicip_id": "2a65923c-7133-415d-ae3b-cf9635a942c5",
          "publicip_address": "10.xx.xx.3",
          "ip_version": 4,
          "publicip_type": "5_bgp"
        }
      ],
      "tenant_id": "26ae5181a416420998eb2093aaed84d9",
      "bandwidth_type": "bgp",
      "status": "NORMAL"
    },
    {
      "id": "0a583ff1-b43e-4000-ade3-e7af0097f832",
      "name": "vpngw-7e880d5b-f458-40ad-a7e5-735c44cd8b7d",
      "size": 300,
      "share_type": "PER",
      "publicip_info": [
        {
          "publicip_id": "c754bc9a-16d5-4763-9674-d7561917aa80",
          "publicip_address": "10.xx.xx.9",
          "ip_version": 4,
          "publicip_type": "5_bgp"
        }
      ],
      "tenant_id": "26ae5181a416420998eb2093aaed84d9",
      "bandwidth_type": "bgp",
      "status": "NORMAL"
    },
    {
      "id": "0a673f00-3640-4a13-949e-7049b2916baf",
      "name": "bandwidth123",
      "size": 10,
      "share_type": "PER",
      "publicip_info": [
        {
          "publicip_id": "cec7fb70-2f82-4561-bd83-2121fb642fdc",
          "publicip_address": "10.xx.xx.184",
          "ip_version": 4,
          "publicip_type": "5_bgp"
        }
      ]
    }
  ]
}
```



```
    ],  
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",  
    "bandwidth_type": "bgp",  
    "status": "NORMAL"  
  },  
  {  
    "id": "0dde1eae-1783-46dc-998c-930fbe261ff9",  
    "name": "bandwidth123",  
    "size": 100,  
    "share_type": "PER",  
    "publicip_info": [  
      {  
        "publicip_id": "24232038-e178-40ad-80e4-5abb75db84be",  
        "publicip_address": "10.xx.xx.101",  
        "ip_version": 4,  
        "publicip_type": "5_bgp"  
      }  
    ],  
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",  
    "bandwidth_type": "bgp",  
    "status": "NORMAL"  
  }  
]
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.2.3 Updating a Bandwidth

Function

This API is used to update information about a bandwidth.

URI

PUT /v1/{project_id}/bandwidths/{bandwidth_id}

[Table 4-27](#) describes the parameters.

Table 4-27 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter

Table 4-28 Request parameter

Name	Mandatory	Type	Description
bandwidth	Yes	Object	Specifies the bandwidth objects. For details, see Table 4-29 .

Table 4-29 Description of the **bandwidth** field

Name	Mandatory	Type	Description
name	No	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.). If the value is left blank, the name of the bandwidth is not changed. Either parameter name or size must be specified.

Name	Mandatory	Type	Description
size	No	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size in Mbit/s. The value ranges from 1 Mbit/s to 300 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.) If the parameter is not included, the bandwidth size is not changed. Either parameter name or size must be specified. If a decimal fraction (for example 10.2) or a character string (for example "10") is specified, the specified value will be automatically converted to an integer. The minimum increment for bandwidth adjustment varies depending on the bandwidth range. The details are as follows: <ul style="list-style-type: none"> The minimum increment is 1 Mbit/s if the allowed bandwidth ranges from 0 Mbit/s to 300 Mbit/s (with 300 Mbit/s included). The minimum increment is 50 Mbit/s if the allowed bandwidth ranges from 300 Mbit/s to 1000 Mbit/s (with 1000 Mbit/s included). The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1000 Mbit/s.

- Example request

PUT https://{Endpoint}/v1/{project_id}/bandwidths/{bandwidth_id}

```
{
  "bandwidth":
    {"name": "bandwidth123",
     "size": 10
    }
}
```

Response Message

- Response parameter

Table 4-30 Response parameter

Name	Type	Description
bandwidth	Object	Specifies the bandwidth objects. For details, see Table 4-31 .

Table 4-31 Description of the **bandwidths** field

Name	Type	Description
name	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).
size	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size in Mbit/s. The value ranges from 1 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	<ul style="list-style-type: none"> The value is PER, indicating that the bandwidth is dedicated.
publicip_info	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies the bandwidth information. For details, see Table 4-32.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	<ul style="list-style-type: none"> Specifies the bandwidth type. The value is bgp.
charge_mode	String	<ul style="list-style-type: none"> Specifies whether the bandwidth is billed by traffic or by bandwidth size. Possible values can be bandwidth (billed by bandwidth) and traffic (billed by traffic). If the value is an empty character string or no value is specified, value bandwidth is used.

Table 4-32 publicip_info objects

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	<ul style="list-style-type: none">Specifies the EIP type.The value can be 5_bgp.The configured value must be supported by the system.

- Example response

```
{
  "bandwidth": {
    "id": "3fa5b383-5a73-4dcb-a314-c6128546d855",
    "name": "bandwidth123",
    "size": 10,
    "share_type": "PER",
    "publicip_info": [
      {
        "publicip_id": "6285e7be-fd9f-497c-bc2d-dd0bdea6efe0",
        "publicip_address": "161.xx.xx.9",
        "publicip_type": "5_bgp",
        "ip_version": 4
      }
    ]
  },
  "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
  "bandwidth_type": "bgp",
  "charge_mode": "bandwidth",

  "status": "NORMAL"
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.3 Bandwidth (V2.0)

4.3.1 Assigning a Shared Bandwidth

Function

This API is used to assign a shared bandwidth.

URI

POST /v2.0/{project_id}/bandwidths

[Table 4-33](#) describes the parameters.

Table 4-33 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .

Request Message

- Request parameter

Table 4-34 Request parameter

Name	Mandatory	Type	Description
bandwidth	Yes	Object	Specifies the bandwidth objects. For details, see Table 4-35 .

Table 4-35 Description of the **bandwidth** field

Name	Mandatory	Type	Description
name	Yes	String	Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).

Name	Mandator y	Type	Description
size	Yes	Integer	<ul style="list-style-type: none"> • Specifies the bandwidth size. The shared bandwidth has a minimum limit, which may vary depending on sites. The default minimum value is 5 Mbit/s. • The value ranges from 5 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.) • If a decimal fraction (for example 10.2) or a character string (for example "10") is specified, the specified value will be automatically converted to an integer. • The minimum increment for bandwidth adjustment varies depending on the bandwidth range. The details are as follows: <ul style="list-style-type: none"> – The minimum increment is 1 Mbit/s if the allowed

Name	Mandator y	Type	Description
			<p>bandwidth ranges from 0 Mbit/s to 300 Mbit/s (with 300 Mbit/s included).</p> <ul style="list-style-type: none"> - The minimum increment is 50 Mbit/s if the allowed bandwidth ranges from 300 Mbit/s to 1000 Mbit/s (with 1000 Mbit/s included). - The minimum increment is 500 Mbit/s if the allowed bandwidth is greater than 1000 Mbit/s.

- Example request
POST https://{Endpoint}/v2.0/{project_id}/bandwidths

```
{
  "bandwidth": {
    "name": "bandwidth123",
    "size": 10
  }
}
```

Response Message

- Response parameter

Table 4-36 Response parameter

Name	Type	Description
bandwidth	Object	Specifies the bandwidth objects. For details, see Table 4-37 .

Table 4-37 Description of the **bandwidth** field

Name	Type	Description
name	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.).
size	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size. The value ranges from 5 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	<ul style="list-style-type: none"> Specifies whether the bandwidth is shared or dedicated. The value can be PER or WHOLE. <ul style="list-style-type: none"> WHOLE: Shared bandwidth PER: Dedicated bandwidth
publicip_info	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies information about the EIP that uses the bandwidth. For details, see Table 4-38. The bandwidth, whose type is WHOLE, can be used by multiple EIPs. The bandwidth, whose type is PER, can be used by only one EIP.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	<ul style="list-style-type: none"> Specifies the bandwidth type. The default value for the shared bandwidth is share.
status	String	<ul style="list-style-type: none"> Specifies the bandwidth status. Possible values are as follows: <ul style="list-style-type: none"> FREEZED (Frozen) NORMAL (Normal)

Table 4-38 publicip_info object

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.
publicip_type	String	<ul style="list-style-type: none">Specifies the EIP type.The value can be 5_bgp.The configured value must be supported by the system.

- Example response

```
{
  "bandwidth": {
    "id": "1bffc5f2-ff19-45a6-96d2-dfdca49cc387",
    "name": "bandwidth123",
    "size": 10,
    "share_type": "WHOLE",
    "publicip_info": [],
    "tenant_id": "26ae5181a416420998eb2093aaed84d9",
    "bandwidth_type": "share",
    "charge_mode": "bandwidth",
    "status": "NORMAL"
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.3.2 Deleting a Shared Bandwidth

Function

This API is used to delete a shared bandwidth.

URI

DELETE /v2.0/{project_id}/bandwidths/{bandwidth_id}

[Table 4-39](#) describes the parameters.

Table 4-39 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth. Currently, only the shared bandwidth can be deleted.

Request Message

- Request parameter
None
- Example request
DELETE https://{Endpoint}/v2.0/{project_id}/bandwidths/{bandwidth_id}

Response Message

- Response parameter
None
- Example response
Or

```
{  
  "code": "xxx",  
  "message": "xxxxx"  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.3.3 Adding an EIP to a Shared Bandwidth

Function

This API is used to add an EIP to a shared bandwidth.

URI

POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/insert

Table 4-40 describes the parameters.

Table 4-40 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter

Table 4-41 Request parameter

Name	Mandatory	Type	Description
bandwidth	Yes	Object	Specifies the bandwidth objects. For details, see Table 4-42 .

Table 4-42 Description of the **bandwidth** field

Name	Mandatory	Type	Description
publicip_info	Yes	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies information about the EIP to be added to the shared bandwidth. For details, see Table 4-43. The bandwidth, whose type is WHOLE, can be used by multiple EIPs. The number of EIPs varies depending on the tenant quota. By default, a shared bandwidth can be used by up to 20 EIPs.

Table 4-43 publicip_info object

Name	Mandatory	Type	Description
publicip_id	Yes	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_type	No	String	<ul style="list-style-type: none"> Specifies the EIP type. Constraints: <ul style="list-style-type: none"> The configured value must be supported by the system. publicip_id is an IPv4 port. If publicip_type is not specified, the default value is 5_bgp.

- Example request

POST `https://{Endpoint}/v2.0/{project_id}/bandwidths/{bandwidth_id}/insert`

```
{
  "bandwidth": {
    "publicip_info": [
      {
        "publicip_type": "5_bgp"
      }
    ]
  }
}
```

Response Message

- Response parameter

Table 4-44 Response parameter

Name	Type	Description
bandwidth	Object	Specifies the bandwidth objects. For details, see Table 4-45 .

Table 4-45 Description of the **bandwidth** field

Name	Type	Description
name	String	<ul style="list-style-type: none"> Specifies the bandwidth name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), hyphens (-), and periods (.)

Name	Type	Description
size	Integer	<ul style="list-style-type: none"> Specifies the bandwidth size. The value ranges from 5 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.)
id	String	Specifies the bandwidth ID, which uniquely identifies the bandwidth.
share_type	String	<ul style="list-style-type: none"> Specifies whether the bandwidth is shared or dedicated. The value can be PER or WHOLE. <ul style="list-style-type: none"> WHOLE: Shared bandwidth PER: Dedicated bandwidth
publicip_info	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies information about the EIP that uses the bandwidth. For details, see Table 4-46. The bandwidth, whose type is WHOLE, can be used by multiple EIPs. The bandwidth, whose type is PER, can be used by only one EIP.
tenant_id	String	Specifies the project ID.
bandwidth_type	String	<ul style="list-style-type: none"> Specifies the bandwidth type. The default value for the shared bandwidth is share.
status	String	<ul style="list-style-type: none"> Specifies the bandwidth status. Possible values are as follows: <ul style="list-style-type: none"> FREEZED (Frozen) NORMAL (Normal)

Table 4-46 publicip_info objects

Name	Type	Description
publicip_id	String	Specifies the ID of the EIP that uses the bandwidth.
publicip_address	String	Specifies the obtained EIP if only IPv4 EIPs are available.

Name	Type	Description
publicip_type	String	<ul style="list-style-type: none">Specifies the EIP type.The configured value must be supported by the system.

- Example response

```
{
  "bandwidth": {
    "id": "3fa5b383-5a73-4dcb-a314-c6128546d855",
    "name": "bandwidth123",
    "size": 10,
    "share_type": "WHOLE",
    "publicip_info": [
      {
        "publicip_id": "1d184b2c-4ec9-49b5-a3f9-27600a76ba3f",
        "publicip_address": "99.xx.xx.82",
        "publicip_type": "5_bgp",
        "ip_version": 4
      }
    ],
    "tenant_id": "8b7e35ad379141fc9df3e178bd64f55c",
    "charge_mode": "traffic",
    "bandwidth_type": "share",
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.3.4 Removing an EIP from a Shared Bandwidth

Function

This API is used to remove an EIP from a shared bandwidth.

URI

POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/remove

[Table 4-47](#) describes the parameters.

Table 4-47 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
bandwidth_id	Yes	Specifies the bandwidth ID, which uniquely identifies the bandwidth.

Request Message

- Request parameter

Table 4-48 Request parameter

Name	Mandatory	Type	Description
bandwidth	Yes	Object	Specifies the bandwidth objects. For details, see Table 4-49 .

Table 4-49 Description of the **bandwidth** field

Name	Mandatory	Type	Description
publicip_info	Yes	Array of publicip_info objects	<ul style="list-style-type: none"> Specifies information about the EIP to be removed from the bandwidth. For details, see Table 4-50. The bandwidth, whose type is WHOLE, can be used by multiple EIPs. The number of EIPs varies depending on the tenant quota. By default, a shared bandwidth can be used by up to 20 EIPs.
charge_mode	Yes	String	<p>After an EIP is removed from a shared bandwidth, a dedicated bandwidth will be allocated to the EIP, and you will be billed for the dedicated bandwidth.</p> <p>Specifies whether the dedicated bandwidth used by the EIP that has been removed from a shared bandwidth is billed by traffic or by bandwidth.</p> <p>The value can be bandwidth or traffic.</p>

Name	Mandatory	Type	Description
size	Yes	Integer	<p>After an EIP is removed from a shared bandwidth, a dedicated bandwidth will be allocated to the EIP, and you will be billed for the dedicated bandwidth.</p> <p>Specifies the size (Mbit/s) of the dedicated bandwidth used by the EIP that has been removed from a shared bandwidth.</p> <p>The value ranges from 1 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can see the available bandwidth range on the management console.)</p>

Table 4-50 publicip_info object

Name	Mandatory	Type	Description
publicip_id	Yes	String	Specifies the ID of the EIP that uses the bandwidth.

- Example request

POST https://{Endpoint}/v2.0/{project_id}/bandwidths/{bandwidth_id}/remove

```
{
  "bandwidth": {
    "publicip_info": [
      {
        "publicip_id": "d91b0028-6f6b-4478-808a-297b75b6812a"
      },
      {
        "publicip_id": "1d184b2c-4ec9-49b5-a3f9-27600a76ba3f"
      }
    ],
    "charge_mode": "traffic",
    "size": 22
  }
}
```

Response Message

- Response parameter

None

- Example response

None

Or

```
{
  "code": "xxx",
  "message": "xxxxx"
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.4 Quota

4.4.1 Querying the Quota

Function

This API is used to query network resource quotas of a tenant. The network resources include VPCs, subnets, security groups, security group rules, EIPs, and VPNs.

NOTE

This API is used to query both the EIP and VPC quotas.

URI

GET /v1/{project_id}/quotas

Example:

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

[Table 4-51](#) describes the parameters.

Table 4-51 Parameter description

Name	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
type	No	String	Specifies the resource type. The value can be vpc , subnet , securityGroup , securityGroupRule , publicip , vpn , vpngw , vpcPeer , firewall , shareBandwidth , shareBandwidthIP , loadbalancer , or listener .

Request Message

- Request parameter
None
- Example request
GET https://{Endpoint}/v1/{project_id}/quotas

Response Message

- Response parameter

Table 4-52 Response parameter

Name	Type	Description
quotas	Object	Specifies the quota object. For details, see Table 4-53 .

Table 4-53 Description of the **quotas** field

Name	Type	Description
resources	Array of resource objects	Specifies the resource objects. For details, see Table 4-54 .

Table 4-54 Description of the **resource** field

Name	Type	Description
type	String	<ul style="list-style-type: none">• Specifies the resource type.• The value can be vpc, subnet, securityGroup, securityGroupRule, publicIp, vpn, vpngw, vpcPeer, firewall, shareBandwidth, or shareBandwidthIP.
used	Integer	<ul style="list-style-type: none">• Specifies the number of created network resources.• The value ranges from 0 to the value of quota.

Name	Type	Description
quota	Integer	<ul style="list-style-type: none"> Specifies the maximum quota values for the resources. The value ranges from the default quota value to the maximum quota value. The default resource quota values can be changed. You must configure the quota values in the underlying system in advance. The default quota values of network resources are as follows: VPC: 5 Subnet: 100 Security group: 100 Security group rule: 5000 EIP: 10 VPN: 5 VPN gateway: 2 VPC peering connection: 50 Firewall: 200 Shared bandwidth: 5 IP address with shared bandwidth: 20
min	Integer	Specifies the minimum quota value allowed.

- Example response

```

{
  "quotas": {
    "resources": [
      {
        "type": "vpc",
        "used": 4,
        "quota": 150,
        "min": 0
      },
      {
        "type": "subnet",
        "used": 5,
        "quota": 400,
        "min": 0
      },
      {
        "type": "securityGroup",
        "used": 1,
        "quota": 100,
        "min": 0
      },
      {
        "type": "securityGroupRule",
        "used": 6,
        "quota": 5000,
        "min": 0
      },
      {
        "type": "publicIp",

```

```
    "used": 2,  
    "quota": 10,  
    "min": 0  
  },  
  {  
    "type": "vpn",  
    "used": 0,  
    "quota": 5,  
    "min": 0  
  },  
  {  
    "type": "vpngw",  
    "used": 0,  
    "quota": 2,  
    "min": 0  
  },  
  {  
    "type": "vpcPeer",  
    "used": 0,  
    "quota": 50,  
    "min": 0  
  },  
  {  
    "type": "firewall",  
    "used": 0,  
    "quota": 200,  
    "min": 0  
  },  
  {  
    "type": "shareBandwidth",  
    "used": 0,  
    "quota": 5,  
    "min": 0  
  },  
  {  
    "type": "shareBandwidthIP",  
    "used": 0,  
    "quota": 20,  
    "min": 0  
  },  
  {  
    "type": "loadbalancer",  
    "used": 0,  
    "quota": 10,  
    "min": 0  
  },  
  {  
    "type": "listener",  
    "used": 0,  
    "quota": 10,  
    "min": 0  
  }  
]  
}
```

Status Codes

See [7.1 Status Codes](#).

Error Codes

See [7.2 Error Codes](#).

4.5 EIP Tag Management

4.5.1 Creating a Tag for an EIP

Function

This API is used to create a tag for an EIP.

URI

POST /v2.0/{project_id}/publicips/{publicip_id}/tags

[Table 4-55](#) describes the parameters.

Table 4-55 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

Table 4-56 Request parameter

Parameter	Type	Mandatory	Description
tag	Object	Yes	Specifies the tag objects. For details, see Table 4-57 .

Table 4-57 tag objects

Attribute	Type	Mandatory	Description
key	String	Yes	<ul style="list-style-type: none">Specifies the tag key.Cannot be left blank.Can contain a maximum of 36 characters.The tag key of a VPC must be unique.
value	String	Yes	<ul style="list-style-type: none">Specifies the tag value.Can contain a maximum of 43 characters.

- Example request

POST `https://{Endpoint}/v2.0/{project_id}/publicips/{publicip_id}/tags`

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

Response Message

- Response parameter

None

- Example response

None

Or

```
{
  "code": "xxx",
  "message": "xxxxx"
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.5.2 Querying EIP Tags

Function

This API is used to query tags of a specified EIP.

URI

GET /v2.0/{project_id}/publicips/{publicip_id}/tags

[Table 4-58](#) describes the parameters.

Table 4-58 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter
None
- Example request
GET https://{Endpoint}/v2.0/{project_id}/publicips/{publicip_id}/tags

Response Message

- Response parameter

Table 4-59 Response parameter

Parameter	Type	Description
tags	Array of tag objects	Specifies the tag object list. For details, see Table 4-60 .

Table 4-60 tag objects

Attribute	Type	Description
key	String	<ul style="list-style-type: none"> Specifies the tag key. Cannot be left blank. Can contain a maximum of 36 characters. The tag key of a VPC must be unique.

Attribute	Type	Description
value	String	<ul style="list-style-type: none"> Specifies the tag value. Can contain a maximum of 43 characters.

- Example response

```
{
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value3"
    }
  ]
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.5.3 Deleting an EIP Tag

Function

This API is used to delete an EIP tag.

URI

DELETE /v2.0/{project_id}/publicips/{publicip_id}/tags/{key}

[Table 4-61](#) describes the parameters.

Table 4-61 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.
key	Yes	Specifies the tag key.

Request Message

- Request parameter
None
- Example request
DELETE https://{Endpoint}/v2.0/{project_id}/publicips/{publicip_id}/tags/{key}

Response Message

- Response parameter
None
 - Example response
None
- Or

```
{  
  "code": "xxx",  
  "message": "xxxxx"  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.5.4 Batch Creating or Deleting EIP Tags

Function

This API is used to add multiple tags to or delete multiple tags from an EIP at a time.

This API is idempotent.

If there are duplicate keys in the request body when you add tags, an error is reported.

During tag creation, duplicate keys are not allowed. If a key already exists in the database, its value will be overwritten by the new duplicate key.

During tag deletion, if some tags do not exist, the operation is considered to be successful by default. The character set of the tags will not be checked. When you delete tags, the tag structure cannot be missing, and the key cannot be left blank or be an empty string.

URI

POST /v2.0/{project_id}/publicips/{publicip_id}/tags/action

[Table 4-62](#) describes the parameters.

Table 4-62 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .
publicip_id	Yes	Specifies the unique identifier of an EIP.

Request Message

- Request parameter

Table 4-63 Request parameter

Parameter	Type	Mandatory	Description
tags	Array of tag objects	Yes	Specifies the tag object list. For details, see Table 4-64 .
action	String	Yes	Specifies the operation. Possible values are as follows: <ul style="list-style-type: none"> create delete

Table 4-64 tag objects

Attribute	Type	Mandatory	Description
key	String	Yes	<ul style="list-style-type: none"> Specifies the tag key. Cannot be left blank. Can contain a maximum of 36 characters. The tag key of a VPC must be unique.

Attribute	Type	Mandatory	Description
value	String	Yes (when the value of action is create) No (when the value of action is delete)	<ul style="list-style-type: none"> Specifies the tag value. Can contain a maximum of 43 characters.

- Request example 1: Creating tags in batches

POST https://{Endpoint}/v2.0/{project_id}/publicips/{publicip_id}/tags/action

```
{
  "action": "create",
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value3"
    }
  ]
}
```

- Request example 2: Deleting tags in batches

POST https://{Endpoint}/v2.0/{project_id}/publicips/{publicip_id}/tags/action

```
{
  "action": "delete",
  "tags": [
    {
      "key": "key1",
      "value": "value1"
    },
    {
      "key": "key2",
      "value": "value3"
    }
  ]
}
```

Response Message

- Response parameter
None

- Example response

None

Or

```
{  
  "code": "xxx",  
  "message": "xxxxx"  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.5.5 Querying EIPs by Tag

Function

This API is used to query EIPs by tag.

URI

POST /v2.0/{project_id}/publicips/resource_instances/action

[Table 4-65](#) describes the parameters.

Table 4-65 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .

Request Message

- Request parameter

Table 4-66 Request parameter

Parameter	Type	Mandatory	Description
tags	Array of tag objects	No	Specifies the included tags. A maximum of tag keys are allowed for each query operation. Each tag key can have up to 10 tag values. The structure body must be included. The tag key cannot be left blank or set to an empty string. Each tag key must be unique, and each tag value in a tag must be unique.
limit	String	No	Sets the page size. This parameter is not available when action is set to count . The default value is 1000 when action is set to filter . The maximum value is 1000 , and the minimum value is 1 . The value cannot be a negative number.
offset	String	No	Specifies the index position. The query starts from the next piece of data indexed by this parameter. This parameter is not required when you query data on the first page. The value in the response returned for querying data on the previous page will be included in this parameter for querying data on subsequent pages. This parameter is not available when action is set to count . If action is set to filter , the value must be a number, and the default value is 0 . The value cannot be a negative number.
action	String	Yes	Specifies the operation to perform. The value can only be filter (filtering) or count (querying the total number). The value filter indicates pagination query. The value count indicates that the total number of query results meeting the search criteria will be returned.
matches	Array of match objects	No	Specifies the search criteria. The tag key is the field to match. Currently, only resource_name is supported. The tag value indicates the matched value. This field is a fixed dictionary value.

Table 4-67 Description of the **tag** field

Name	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The value can contain a maximum of 127 Unicode characters. The tag key cannot be left blank. (This parameter is not verified during the search process.)
values	Yes	Array of strings	Specifies the tag value list. Each value can contain a maximum of 255 Unicode characters. An empty list for values indicates any value. The values are in the OR relationship.

Table 4-68 Description of the **match** field

Name	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. Currently, the tag key can only be the resource name.
value	Yes	String	Specifies the tag value. A value contains a maximum of 255 Unicode characters.

- Example request 1: Setting **action** to **filter**

POST https://{Endpoint}/v2.0/{project_id}/publicips/resource_instances/action

```
{
  "offset": "0",
  "limit": "100",
  "action": "filter",
  "matches": [
    {
      "key": "resource_name",
      "value": "resource1"
    }
  ],
  "tags": [
    {
      "key": "key1",
      "values": [
        "*value1",
        "value2"
      ]
    }
  ]
}
```

- Example request 2: Setting **action** to **count**

```
{
  "action": "count",
  "tags": [
    {
      "key": "key1",
      "values": [
        "value1",
        "value2"
      ]
    },
    {
      "key": "key2",
      "values": [
        "value1",
        "value2"
      ]
    }
  ],
  "matches": [
    {
      "key": "resource_name",
      "value": "resource1"
    }
  ]
}
```

Response Message

- Response parameter

Table 4-69 Response parameter

Name	Type	Description
resources	Array of resource objects	Specifies the resource object list. For details, see Table 4-70 .
total_count	Integer	Specifies the total number of query records.

Table 4-70 resource objects

Name	Type	Description
resource_id	String	Specifies the resource ID.
resource_detail	Object	Specifies the resource details. Resource details are used for extension. This parameter is left blank by default.

Name	Type	Description
tags	Array of tag objects	Specifies the tag list. This parameter is an empty array by default if there is no tag. For details, see Table 4-71 .
resource_name	String	Specifies the resource name. This parameter is an empty string by default if there is no resource name.

Table 4-71 Description of the **tag** field

Name	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The value can contain a maximum of 127 Unicode characters. The tag key cannot be left blank. (This parameter is not verified during the search process.)
values	Yes	Array of strings	Specifies the tag value list. Each value can contain a maximum of 255 Unicode characters. An empty list for values indicates any value. The values are in the OR relationship.

- Example response 1: Setting **action** to **filter**

```

{
  "resources": [
    {
      "resource_detail": null,
      "resource_id": "cdfs_cefs_wesas_12_dsad",
      "resource_name": "resouece1",
      "tags": [
        {
          "key": "key1",
          "value": "value1"
        },
        {
          "key": "key2",
          "value": "value1"
        }
      ]
    }
  ],
  "total_count": 1000
}

```

- Example response 2: Setting **action** to **count**

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

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

4.5.6 Querying EIP Tags in a Specified Project

Function

This API is used to query all EIP tags of a tenant in a specified region.

URI

GET /v2.0/{project_id}/publicips/tags

[Table 4-72](#) describes the parameters.

Table 4-72 Parameter description

Name	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain a project ID, see A.2 Obtaining a Project ID .

Request Message

- Request parameter
None
- Example request
GET /v2.0/{project_id}/publicips/tags

Response Message

- Response parameter

Table 4-73 Response parameter

Parameter	Type	Description
tags	Array of tag objects	Specifies the tag object list. For details, see Table 4-74 .

Table 4-74 Description of the **tag** field

Name	Type	Description
key	String	Specifies the tag key. <ul style="list-style-type: none">• Cannot be left blank.• Can contain a maximum of 36 characters.
values	Array of strings	Specifies the tag value list. <ul style="list-style-type: none">• Can contain a maximum of 43 characters.

- Example response

```
{
  "tags": [
    {
      "key": "key1",
      "values": [
        "value1",
        "value2"
      ]
    },
    {
      "key": "key2",
      "values": [
        "value1",
        "value2"
      ]
    }
  ]
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5 Native OpenStack Neutron APIs V2.0

5.1 API Version Information

5.1.1 Querying API Versions

Function

This API is used to query all available versions of a native OpenStack API.

URI

GET /

Request Message

Request parameter

None

Example request

```
GET https://{Endpoint}/
```

Response Message

Response parameter

Table 5-1 Response parameter

Parameter	Type	Description
versions	Array of version objects	Specifies the API version list. For details, see Table 5-2 .

Table 5-2 version objects

Parameter	Type	Description
status	String	Specifies the API version status. Possible values are as follows: <ul style="list-style-type: none">• CURRENT• STABLE• DEPRECATED
id	String	Specifies the API version.
links	Array of link objects	Specifies the link list. For details, see Table 5-3 .

Table 5-3 link objects

Parameter	Type	Description
href	String	Specifies the API link.
rel	String	Specifies the relationship between the API link and the API version.

Example response

```
{
  "versions": [
    {
      "status": "CURRENT",
      "id": "v2.0",
      "links": [
        {
          "href": "https://None/v2.0",
          "rel": "self"
        }
      ]
    }
  ]
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.1.2 Pagination

Function

Neutron APIs v2.0 provides the pagination function. You can set parameters **limit** and **marker** in the URL to enable the desired number of items to be returned. All returned items are displayed in the ascending order of ID.

- To access the next page of the request, perform the following configurations:
 - Replace the value of **marker** in the original access request URL. Replace the value of **marker** to the value of **marker** in the value of **herf** if the value of **rel** in the response is **next**.
 - Set the value of **page_reverse** to **False**.
- To access the previous page of the request, perform the following configurations:
 - Replace the value of **marker** in the original access request URL. Replace the value of **marker** to the value of **marker** in the value of **herf** if the value of **rel** in the response is **previous**.
 - Set the value of **page_reverse** to **True**.

Request Message

Request parameter

Table 5-4 Request parameter

Parameter	Type	Mandatory	Description
limit	String	No	Specifies the number of items displayed per page.
marker	String	No	Specifies the ID of the last item in the previous list. If the marker value is invalid, error code 400 will be returned.
page_reverse	Boolean	No	Specifies the page direction. The value can be True or False .

Example request 1

```
GET https://{Endpoint}/v2.0/networks?limit=2&marker=3d42a0d4-a980-4613-ae76-a2cddecff054&page_reverse=False
```

Example request 2

```
GET https://{Endpoint}/v2.0/vpc/peerings?limit=2&marker=e5a0c88e-228e-4e62-a8b0-90825b1b7958&page_reverse=True
```

Response Message

Response parameter

None

Example response 1

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [],
      "name": "liudongtest ",
      "admin_state_up": false,
      "tenant_id": "6fbe9263116a4b68818cf1edce16bc4f",
      "id": "60c809cb-6731-45d0-ace8-3bf5626421a9"
    },
    {
      "status": "ACTIVE",
      "subnets": [
        "132dc12d-c02a-4c90-9cd5-c31669aace04"
      ],
      "name": "publicnet",
      "admin_state_up": true,
      "tenant_id": "6fbe9263116a4b68818cf1edce16bc4f",
      "id": "9daeac7c-a98f-430f-8e38-67f9c044e299"
    }
  ],
  "networks_links": [
    {
      "href": "http://192.168.82.231:9696/v2.0/networks?limit=2&marker=9daeac7c-a98f-430f-8e38-67f9c044e299",
      "rel": "next"
    },
    {
      "href": "http://192.168.82.231:9696/v2.0/networks?limit=2&marker=60c809cb-6731-45d0-ace8-3bf5626421a9&page_reverse=True",
      "rel": "previous"
    }
  ]
}
```

Example response 2

```
{
  "peerings_links": [
    {
      "marker": "dd442819-5638-401c-bd48-a82703cf0464",
      "rel": "next"
    },
    {
      "marker": "1e13cbaf-3ce4-413d-941f-66d855dbfa7f",
      "rel": "previous"
    }
  ],
  "peerings": [
    {
      "status": "ACTIVE",
      "accept_vpc_info": {
        "vpc_id": "83a48834-b9bc-4f70-aa46-074568594650",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "request_vpc_info": {
        "vpc_id": "db8e7687-e43b-4fc1-94cf-16f69f484d6d",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      },
      "name": "peering1",
      "id": "1e13cbaf-3ce4-413d-941f-66d855dbfa7f"
    },
    {
      "status": "ACTIVE",
      "accept_vpc_info": {
        "vpc_id": "83a48834-b9bc-4f70-aa46-074568594650",
        "tenant_id": "e41a43bf06e249678413c6d61536eff9"
      }
    }
  ]
}
```

```
"request_vpc_info": {  
  "vpc_id": "bd63cc9e-e7b8-4d4e-a0e9-055031470ffc",  
  "tenant_id": "e41a43bf06e249678413c6d61536eff9"  
},  
"name": "peering2",  
"id": "dd442819-5638-401c-bd48-a82703cf0464"  
}  
]  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.2 Floating IP Address

5.2.1 Querying Floating IP Addresses

Function

This API is used to query all floating IP addresses accessible to the tenant submitting the request.

You can query the detailed information about a specified floating IP address using the API for [Querying a Floating IP Address](#).

URI

GET /v2.0/floatingips

[Table 5-5](#) describes the parameters.

Table 5-5 Parameter description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the floating IP address ID.
floating_ip_addresses	No	String	Specifies the floating IPv6 address.

Parameter	Mandatory	Type	Description
floating_network_id	No	String	Specifies the external network ID. You can only use fixed external network. You can use GET /v2.0/networks?router:external=True or GET /v2.0/networks?name={floating_network} or run the neutron net-external-list command to obtain information about the external network.
router_id	No	String	Specifies the ID of the belonged router.
port_id	No	String	Specifies the port ID.
fixed_ip_address	No	String	Specifies the private IP address of the associated port.
tenant_id	No	String	Specifies the project ID.
limit	String	No	Specifies the number of items displayed on each page.
marker	String	No	Specifies the ID of the last item on the previous page. If the marker value is invalid, error code 400 will be returned.

Parameter	Mandatory	Type	Description
page_reverse	Boolean	No	Specifies the page direction. The value can be True or False .

Example:

```
GET https://{Endpoint}/v2.0/floatingips?
id={fip_id}&router_id={router_id}&floating_network_id={net_id}&floating_ip_address={floating_ip}&port_id={
port_id}&fixed_ip_address={fixed_ip}&tenant_id={tenant_id}
```

Request Message

None

Response Message

Table 5-6 Response parameter

Parameter	Type	Description
floatingips	Array of floatingip objects	Specifies the floating IP address list. For details, see Table 5-7 .

Table 5-7 floatingip objects

Parameter	Type	Description
status	String	Specifies the floating IP address status. The value can be ACTIVE , DOWN , or ERROR . <ul style="list-style-type: none"> • DOWN indicates that the floating IP address has not been bound. • ACTIVE indicates that the floating IP address has been bound. • ERROR indicates that the floating IP address is abnormal.

Parameter	Type	Description
id	String	Specifies the floating IP address ID.
project_id	String	Specifies the project ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.
created_at	String	Specifies the time when the floating IP address was created. UTC time is used. Format: <i>yyyy-MM-ddTHH:mm:ss</i>
updated_at	String	Specifies the time when the floating IP address was updated. UTC time is used. Format: <i>yyyy-MM-ddTHH:mm:ss</i>

Example:

Example request

```
GET https://{Endpoint}/v2.0/floatingips?limit=1
```

Example response

```
{
  "floatingips": [
    {
      "id": "1a3a2818-d9b4-4a9c-8a19-5252c499d1cd",
      "status": "DOWN",
      "router_id": null,
      "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
      "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
      "fixed_ip_address": null,
    }
  ]
}
```

```
"floating_ip_address": "99.99.99.84",  
"port_id": null,  
"created_at": "2017-10-19T12:21:28",  
"updated_at": "2018-07-30T12:52:13"  
}  
]  
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.2.2 Querying a Floating IP Address

Function

This API is used to query details about a specified floating IP address, including the floating IP address status, ID of the router to which the floating IP address belongs, and external network ID of the floating IP address.

URI

GET /v2.0/floatingips/{floatingip_id}

Request Message

None

Response Message

Table 5-8 Response parameter

Parameter	Type	Description
floatingip	Object	Specifies the floating IP address list. For details, see Table 5-9 .

Table 5-9 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be ACTIVE , DOWN , or ERROR . <ul style="list-style-type: none"> • DOWN indicates that the floating IP address has not been bound. • ACTIVE indicates that the floating IP address has been bound. • ERROR indicates that the floating IP address is abnormal.
id	String	Specifies the floating IP address ID.
project_id	String	Specifies the project ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.
created_at	String	Specifies the time when the floating IP address was created. UTC time is used. Format: <i>yyyy-MM-ddTHH:mm:ss</i>

Attribute	Type	Description
updated_at	String	Specifies the time when the floating IP address was updated. UTC time is used. Format: <i>yyyy-MM-ddTHH:mm:ss</i>

Example:

Example request

```
GET https://{Endpoint}/v2.0/floatingips/1a3a2818-d9b4-4a9c-8a19-5252c499d1cd
```

Example response

```
{
  "floatingip": {
    "id": "1a3a2818-d9b4-4a9c-8a19-5252c499d1cd",
    "status": "DOWN",
    "router_id": null,
    "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
    "project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
    "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
    "fixed_ip_address": null,
    "floating_ip_address": "99.99.99.84",
    "port_id": null,
    "created_at": "2017-10-19T12:21:28",
    "updated_at": "2018-07-30T12:52:13"
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.2.3 Assigning a Floating IP Address

Function

When assigning a floating IP address, you need to obtain the external network ID **floating_network_id** of the floating IP address.

You can use **GET /v2.0/networks?router:external=True** or run the **neutron net-external-list** command to obtain the UUID of the external network required for assigning a floating IP address.

URI

POST /v2.0/floatingips

Request Message

Table 5-10 Request parameter

Parameter	Type	Mandatory	Description
floatingip	Object	Yes	Specifies the floating IP address list. For details, see Table 5-11 .

Table 5-11 floatingip objects

Parameter	Mandatory	Type	Description
floating_ip_address	No	String	Specifies the floating IP address.
floating_network_id	Yes	String	Specifies the external network ID. You can only use fixed external network. You can use GET /v2.0/networks?router:external=True or GET /v2.0/networks?name={floating_network} or run the neutron net-external-list mode command to obtain information about the external network.
port_id	No	String	Specifies the port ID.
fixed_ip_address	No	String	Specifies the private IP address of the associated port.

Response Message

Table 5-12 Response parameter

Parameter	Type	Description
floatingip	Object	Specifies the floating IP address list. For details, see Table 5-13 .

Table 5-13 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be ACTIVE , DOWN , or ERROR . <ul style="list-style-type: none"> • DOWN indicates that the floating IP address has not been bound. • ACTIVE indicates that the floating IP address has been bound. • ERROR indicates that the floating IP address is abnormal.
id	String	Specifies the floating IP address ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.

Example:

Example request

```
POST https://{Endpoint}/v2.0/floatingips
{
  "floatingip": {
    "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
  }
}
```

Example response

```
{
  "floatingip": {
```



```
{
  "id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
  "status": "DOWN",
  "router_id": null,
  "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
  "project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
  "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
  "fixed_ip_address": null,
  "floating_ip_address": "88.88.215.205",
  "port_id": null,
  "created_at": "2018-09-20T02:10:02",
  "updated_at": "2018-09-20T02:10:02"
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.2.4 Updating a Floating IP Address

Function

This API is used to update a floating IP address.

During the update, the ID of the floating IP address must be provided in the URL.

If **port_id** is left blank, the floating IP address has been unbound from the port.

Restrictions

When you bind a floating IP address, if the floating IP address is in the **error** state, try unbinding the address first.

You are not allowed to associate a port that has a floating IP address associated to another floating IP address. You must first disassociate the port from the IP address and then associate it with another IP address.

URI

PUT /v2.0/floatingips/{floatingip_id}

[Table 5-14](#) describes the parameters.

Table 5-14 Parameter description

Parameter	Mandatory	Type	Description
floatingip_id	Yes	String	Specifies the floating IP address ID. This parameter is not required when you assign a floating IP address. This parameter is mandatory when you query, update, or delete a floating IP address.

Request Message

Table 5-15 Request parameter

Parameter	Type	Mandatory	Description
floatingip	Object	Yes	Specifies the floating IP address list. For details, see Table 5-16 .

Table 5-16 floatingip objects

Parameter	Mandatory	Type	Description
port_id	No	String	Specifies the port ID.

Response Message

Table 5-17 Response parameter

Parameter	Type	Description
floatingip	Object	Specifies the floating IP address list. For details, see Table 5-18 .

Table 5-18 floatingip objects

Attribute	Type	Description
status	String	Specifies the floating IP address status. The value can be ACTIVE , DOWN , or ERROR . <ul style="list-style-type: none"> • DOWN indicates that the floating IP address has not been bound. • ACTIVE indicates that the floating IP address has been bound. • ERROR indicates that the floating IP address is abnormal.
id	String	Specifies the floating IP address ID.
floating_ip_address	String	Specifies the floating IP address.
floating_network_id	String	Specifies the external network ID.
router_id	String	Specifies the ID of the belonged router.
port_id	String	Specifies the port ID.
fixed_ip_address	String	Specifies the private IP address of the associated port.
tenant_id	String	Specifies the project ID.

Example:

Example request 1 (Binding a floating IP address to a port)

PUT https://{Endpoint}/v2.0/floatingips/b997e0d4-3359-4c74-8f88-bc0af81cd5a2

```
{
  "floatingip": {
    "port_id": "f91f5763-c5a2-4458-979d-61e48b3c3fac"
  }
}
```

Example response 1 (Binding a floating IP address to a port)

```
{
  "floatingip": {
```

```
"id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
"status": "DOWN",
"router_id": null,
"tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
"project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
"floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
"fixed_ip_address": "192.168.10.3",
"floating_ip_address": "88.88.215.205",
"port_id": "00587256-27e3-489b-96bf-ea80c1da4aeb",
"created_at": "2018-09-20T02:10:02",
"updated_at": "2018-09-20T02:10:07"
}
}
```

Example request 2 (Unbinding a floating IP address from a port)

```
PUT https://{Endpoint}/v2.0/floatingips/b997e0d4-3359-4c74-8f88-bc0af81cd5a2
{
  "floatingip": {
    "port_id": null
  }
}
```

Example response 2 (Unbinding a floating IP address from a port)

```
{
  "floatingip": {
    "id": "b997e0d4-3359-4c74-8f88-bc0af81cd5a2",
    "status": "DOWN",
    "router_id": null,
    "tenant_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
    "project_id": "bbfe8c41dd034a07bebd592bf03b4b0c",
    "floating_network_id": "0a2228f2-7f8a-45f1-8e09-9039e1d09975",
    "fixed_ip_address": null,
    "floating_ip_address": "88.88.215.205",
    "port_id": null,
    "created_at": "2018-09-20T02:10:02",
    "updated_at": "2018-09-20T02:10:07"
  }
}
```

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

5.2.5 Deleting a Floating IP Address

Function

This API is used to delete a floating IP address.

URI

```
DELETE /v2.0/floatingips/{floatingip_id}
```

[Table 5-19](#) describes the parameters.

Table 5-19 Parameter description

Parameter	Mandatory	Type	Description
floatingip_id	Yes	String	Specifies the floating IP address ID.

Request Message

None

Response Message

None

Example:

Example request

```
DELETE https://{Endpoint}/v2.0/floatingips/a95ec431-8473-463b-aede-34fb048ee3a7
```

Example response

None

Status Code

See [7.1 Status Codes](#).

Error Code

See [7.2 Error Codes](#).

6 Permissions Policies and Supported Actions

6.1 Elastic IP

Permission	API	Action
Assigns an EIP.	POST /v1/{project_id}/publicips	vpc:publicips:create
Queries an EIP.	GET /v1/{project_id}/publicips/{publicip_id}	vpc:publicips:get
Queries EIPs.	GET /v1/{project_id}/publicips	vpc:publicips:list
Updates an EIP.	PUT /v1/{project_id}/publicips/{publicip_id}	vpc:publicips:update
Release an EIP.	DELETE /v1/{project_id}/publicips/{publicip_id}	vpc:publicips:delete

6.2 Bandwidth

Permission	API	Action
Queries a bandwidth.	GET /v1/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:get
Queries bandwidths.	GET /v1/{project_id}/bandwidths	vpc:bandwidths:list
Updates a bandwidth.	PUT /v1/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:update

6.3 Bandwidth (V2.0)

Permission	API	Action
Allocates a shared bandwidth.	POST /v2.0/{project_id}/bandwidths	vpc:bandwidths:create
Deletes a shared bandwidth.	DELETE /v2.0/{project_id}/bandwidths/{bandwidth_id}	vpc:bandwidths:delete
Adds an EIP to a shared bandwidth.	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/insert	vpc:publicIps:insert
Removes an EIP from a shared bandwidth.	POST /v2.0/{project_id}/bandwidths/{bandwidth_id}/remove	vpc:publicIps:remove

6.4 EIP Tags

Permission	API	Action
Creating a Tag for an EIP	POST /v2.0/{project_id}/publicIps/{publicIp_id}/tags	vpc:publicIpTags:create
Querying EIP Tags	GET /v2.0/{project_id}/publicIps/{publicIp_id}/tags	vpc:publicIpTags:get
Deleting an EIP Tag	DELETE /v2.0/{project_id}/publicIps/{publicIp_id}/tags/{key}	vpc:publicIpTags:delete
Batch Creating or Deleting EIP Tags	POST /v2.0/{project_id}/publicIps/{publicIp_id}/tags/action	vpc:publicIpTags:create vpc:publicIpTags:delete
Querying EIPs by Tag	POST /v2.0/{project_id}/publicIps/resource_instances/action	vpc:publicIpTags:get
Querying EIP Tags in a Specified Project	GET /v2.0/{project_id}/publicIps/tags	vpc:publicIpTags:get

6.5 Floating IP Address (OpenStack Neutron API)

Permission	API	Action
Queries floating IP addresses.	GET /v2.0/floatingips	vpc:floatingips:get
Queries a floating IP address.	GET /v2.0/floatingips/{floatingip_id}	vpc:floatingips:get
Creates a floating IP address.	POST /v2.0/floatingips	vpc:floatingips:create
Updates a floating IP address.	PUT /v2.0/floatingips/{floatingip_id}	vpc:floatingips:update
Deletes a floating IP address.	DELETE /v2.0/floatingips/{floatingip_id}	vpc:floatingips:delete

6.6 Precautions for API Permissions

If you have insufficient permissions, response code **200** will be returned when you query network resources and an empty list will be displayed.

7 Public Parameters

7.1 Status Codes

Table 7-1 Normal values

Normal Response Code	Type	Description
200	OK	Specifies the normal response code for the GET, PUT, and POST operations.
201	Created	Specifies the normal response code for the POST operation of the OpenStack Neutron API.
204	No Content	Specifies the normal response code for the DELETE operation.

Table 7-2 Abnormal values

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

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

7.2 Error Codes

Description

If an error occurs when an API is called, error information is returned. This section describes the error information for EIP APIs (excluding native OpenStack APIs).

Example of Returned Error Information

```
{  
  "code": "VPC.0504",  
  "message": "Floating IP could not be found."  
}
```

Error Code Description

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
Public	400	VPC.0002	The AZ is left blank.	Available zone Name is null.	Check whether the availability_zone field in the request body for creating a subnet is left blank.
	404	VPC.0003	The VPC does not exist.	VPC does not exist.	Check whether the VPC ID is correct or whether the VPC exists under the tenant.
	400	VPC.0004	The VPC status is abnormal.	VPC is not active, please try later.	Try again later or contact technical support.
	401	VPC.0009	Real-name authentication fails.	real-name authentication fail.	Contact technical support.
Public	400	VPC.0007	Inconsistent tenant IDs.	urlTenantId is not equal tokenTenantId	The tenant ID in the URL is different from that parsed in the token.
	401	VPC.0008	Invalid token.	Invalid token in the header.	Check whether the token in the request header is valid.
	403	VPC.2701	You do not have permission to perform this operation, or your account balance is insufficient.	Token not allowed to do this action.	Check whether the account balance is insufficient or whether your account has been frozen.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
Assigning an EIP	400	VPC.0301	The specified bandwidth parameter for assigning an EIP is invalid.	Bandwidth name or share_type is invalid.	Check whether the specified bandwidth parameter is valid.
	400	VPC.0501	Invalid EIP parameters.	Bandwidth share_type is invalid.	Check whether the parameter values are valid based on the returned error message and API reference document.
	403	VPC.0502	You are not allowed to assign the EIP.	Tenant status is op_restricted.	Check whether the account balance is insufficient or whether your account has been frozen.
	500	VPC.0503	Failed to assign the EIP.	Creating publicip failed.	Contact technical support.
	500	VPC.0504	Failed to assign the EIP because no IP address is found.	FloatIp is null.	Contact technical support.
	500	VPC.0508	Port-related resources could not be found.	Port is invalid.	Contact technical support.
	409	VPC.0510	The EIP has already been bound to another ECS.	Floatingip has already associated with port.	Unbind the EIP from the ECS.
	409	VPC.0511	The port has already been bound with an EIP.	Port has already associated with floatingip.	Disassociate the port from the EIP.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
	409	VPC.0521	Insufficient EIP quota.	Quota exceeded for resources : ['floating ip'].	Release the unbound EIPs or request to increase the EIP quota.
	409	VPC.0522	The IP address is invalid or in use.	The IP address is in use.	Check whether the IP address format is valid or replace it with another IP address.
	409	VPC.0532	Failed to assign the IP address because no IP addresses are available.	No more IP addresses available on network.	Release unbound EIPs or try again later.
Querying an EIP	400	VPC.0501	Invalid EIP parameters.	Invalid floatingip_id.	Check whether the EIP ID is valid.
	404	VPC.0504	The EIP could not be found.	Floating IP could not be found.	Check whether the specified EIP ID is valid.
	500	VPC.0514	An exception occurs in the IaaS OpenStack system.	Neutron Error.	Check whether the Neutron service is normal or contact technical support.
Querying EIPs	400	VPC.0501	Invalid EIP parameters.	Invalid limit.	Check whether the parameter values are valid based on the returned error message and API reference document.
Releasing an EIP	400	VPC.0501	Invalid EIP parameters.	Invalid param.	Contact technical support.
	404	VPC.0504	The EIP could not be found.	Floating IP could not be found.	Check whether the specified EIP ID is valid.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
	409	VPC.0512	The EIP status is abnormal.	Resource status is busy, try it again later.	Try again later or contact technical support.
	500	VPC.0513	Network resources cannot be found.	getElementByKey error.	Contact technical support.
	500	VPC.0516	Failed to release the EIP because it is being used by a load balancer.	Publicip is in used by ELB.	Unbind the EIP from the load balancer.
	409	VPC.0517	Failed to release the EIP because it is bound to an ECS.	Floating ip has associated with port, please disassociate it firstly.	Unbind the EIP from the ECS.
	500	VPC.0518	Failed to release the EIP because it is being used by a network ACL.	Public IP has firewall rules.	Contact technical support.
Updating an EIP	400	VPC.0501	Invalid EIP parameters.	Port id is invalid.	Check whether the port ID is valid.
	404	VPC.0504	The EIP could not be found.	Floating IP could not be found.	Check whether the specified EIP ID is valid.
	500	VPC.0509	The port has already been associated with an EIP.	Floating ip double status is invalid.	Disassociate the port from the EIP.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
	409	VPC.0510	The EIP has already been bound to another ECS.	Floatingip has already associated with port.	Unbind the EIP from the ECS.
	409	VPC.0511	Failed to bind the EIP to the ECS because another EIP has already been bound to the ECS.	Port has already associated with floatingip.	Unbind the EIP from the ECS.
	409	VPC.0512	The EIP status is abnormal.	Resource status is busy, try it again later.	Try again later or contact technical support.
	404/500	VPC.0514	An exception occurs in the IaaS OpenStack system.	Neutron Error.	Check whether the Neutron service is normal or contact technical support.
Querying a bandwidth	400	VPC.0301	The bandwidth parameters are incorrect.	getBandwidth error bandwidthid is invalid.	Check whether the bandwidth ID is valid.
	404	VPC.0306	The bandwidth object does not exist.	No Eip bandwidth exist with id.	The bandwidth object to be queried does not exist.
	500	VPC.0302	An exception occurs in the IaaS OpenStack system.	Neutron Error.	Check whether the Neutron service is normal or contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
Querying bandwidths	400	VPC.0301	The bandwidth parameters are incorrect.	Get bandwidths error limit is invalid.	Check whether the parameter values are valid based on the returned error message and API reference document.
	404	VPC.0306	The bandwidth object does not exist.	No Eip bandwidth exist with id.	The bandwidth object to be queried does not exist.
	500	VPC.0302	An exception occurs in the IaaS OpenStack system.	Neutron Error.	Check whether the Neutron service is normal or contact technical support.
Updating a bandwidth	400	VPC.0301	The bandwidth parameters are incorrect.	updateBandwidth input param is invalid.	Check whether the parameter values are valid based on the returned error message and API reference document.
	500	VPC.0302	Failed to obtain underlying resources.	Neutron Error.	Check whether the Neutron service is normal or contact technical support.
	500	VPC.0305	An internal error occurs during the bandwidth update.	updateBandwidth error.	Contact technical support.
Assigning a shared bandwidth	400	VPC.0310	Insufficient shared bandwidth quota.	NO QUOTAS for shareBandwidth!	Delete the shared bandwidth that is no longer required or contact technical support.
Adding an EIP to or removing an EIP from a shared	400	VPC.0301	Invalid EIP.	Invalid publicip_id	Check whether the value of publicip_id in publicip_info is valid.

Module	HTTP Status Code	Error Code	Description	Error Message	Handling Measure
bandwidth	400	VPC.0323	Failed to add an EIP to or remove an EIP from a shared bandwidth.	publicip can not be operate with this bandwidth	Check whether the shared bandwidth or EIP is normal.
Querying the Quota	400	VPC.1207	The specified resource type does not exist.	resource type is invalid.	Use an existing resource type.

A Appendix

A.1 VPC Monitoring Metrics

Description

This section describes monitoring metrics reported by VPC to Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics of the monitored object and alarms generated for VPC.

Namespace

SYS.VPC

Metrics

Table A-1 EIP and Bandwidth metrics

Metric	Metric Name	Description	Value Range	Measurement Object & Dimension	Monitoring Interval (Raw Data)
upstream_bandwidth	Outbound Bandwidth	Network rate of outbound traffic (Previously called "Upstream Bandwidth") Unit: bit/s	≥ 0 bit/s	Object: Bandwidth or EIP Dimension ^a : bandwidth_id, publicip_id	1 minute

Metric	Metric Name	Description	Value Range	Measurement Object & Dimension	Monitoring Interval (Raw Data)
downstream_bandwidth	Inbound Bandwidth	Network rate of inbound traffic (Previously called "Downstream Bandwidth") Unit: bit/s	≥ 0 bit/s	Object: Bandwidth or EIP Dimension: bandwidth_id, publicip_id	1 minute
upstream_bandwidth_usage	Outbound Bandwidth Usage	Usage of outbound bandwidth in the unit of percent.	0% to 100%	Object: Bandwidth or EIP Dimension: bandwidth_id, publicip_id	1 minute
upstream	Outbound Traffic	Network traffic going out of the cloud platform (Previously called "Upstream Traffic") Unit: byte	≥ 0 bytes	Object: Bandwidth or EIP Dimension: bandwidth_id, publicip_id	1 minute
downstream	Inbound Traffic	Network traffic going into the cloud platform (Previously called "Downstream Traffic") Unit: byte	≥ 0 bytes	Object: Bandwidth or EIP Dimension: bandwidth_id, publicip_id	1 minute

Metric	Metric Name	Description	Value Range	Measurement Object & Dimension	Monitoring Interval (Raw Data)
<p>a: If a service has multiple dimensions, all dimensions are mandatory when you use APIs to query the metrics.</p> <ul style="list-style-type: none"> Query a monitoring metric: dim.0=bandwidth_id,530cd6b0-86d7-4818-837f-935f6a27414d&dim.1=publicip_id,3773b058-5b4f-4366-9035-9bbd9964714a Query monitoring metrics in batches: "dimensions": [<pre>{ "name": "bandwidth_id", "value": "530cd6b0-86d7-4818-837f-935f6a27414d" } { "name": "publicip_id", "value": "3773b058-5b4f-4366-9035-9bbd9964714a" }],</pre> 					

Dimension

Key	Value
publicip_id	Specifies the EIP ID.
bandwidth_id	Specifies the bandwidth ID.

A.2 Obtaining a Project ID

Scenarios

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

- [Obtain the Project ID by Calling an API](#)
- [Obtain the Project ID from the Console](#)

Obtain the Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is GET `https://{Endpoint}/v3/projects`. `{Endpoint}` is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [3.2 Authentication](#).

The following is an example response. The value of `id` is the project ID.

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

Obtain a Project ID from the Console

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

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

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

B Change History

Released On	Description
2020-11-05	This issue is the first official release.