NAT Gateway

API Reference

Issue 01

Date 2022-04-12

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Before You Start

1.1 Overview

Welcome to *NAT Gateway API Reference*. The NAT Gateway service provides the network address translation (NAT) function for Elastic Cloud Servers (ECSs) and Bare Metal Servers (BMSs) in a Virtual Private Cloud (VPC). It also supports this function for servers that connect to a VPC through Direct Connect or Virtual Private Network (VPN) in on-premises data centers. It allows these servers to access the Internet using elastic IP addresses (EIPs) or to provide services accessible from the Internet.

This document describes how to use application programming interfaces (APIs) to perform operations on NAT gateways, such as creating or deleting NAT gateways, or adding SNAT rules. For details about all supported operations, see **API Overview**.

1.2 API Calling

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

1.3 Endpoints

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

1.4 Notes and Constraints

- For more details, see the constraints described in each API.

1.5 Concepts

- Account

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

- User

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

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

- Region

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

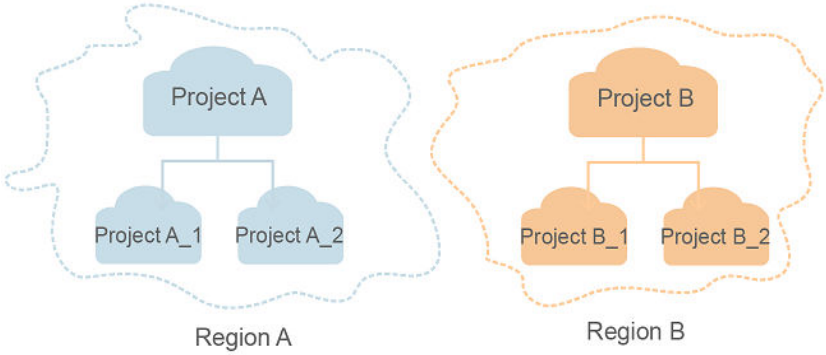
- AZ

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

- Project

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

  Figure 1-1 Project isolation model



- Enterprise project

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

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

1.6 Selecting an API Type

NAT Gateway API can be of v2.0 and v2. v2 APIs are of a later version and will be continuously optimized. You are advised to use v2 APIs.

2 API Overview

NAT Gateway APIs allow you to use all NAT Gateway functions.

API v2

Table 2-1 NAT gateway API v2

Type	Description
NAT Gateway Service	Create, query, update, and delete NAT gateways, including creating a NAT gateway, querying NAT gateways, querying details about a NAT gateway, updating a NAT gateway, and deleting a NAT gateway.
SNAT Rules	Create, query, update, and delete SNAT rules, including creating an SNAT rule, querying SNAT rules, querying details about an SNAT rule, updating an SNAT rule, and deleting an SNAT rule.
DNAT Rules	Create, query, update, and delete DNAT rules, including creating a DNAT rule, creating DNAT rules in batches, querying DNAT rules, querying details about a DNAT rule, updating a DNAT rule, and deleting a DNAT rule.

Table 2-2 NAT gateway API (v2) description

Type	API	Description
NAT Gateway Service	Creating a NAT Gateway	This API is used to create a NAT gateway.
	Querying NAT Gateways	This API is used to query NAT gateways.
	Querying Details About a Specified NAT Gateway	This API is used to query details about a NAT gateway.

Type	API	Description
	Updating a NAT Gateway	This API is used to update a NAT gateway.
	Deleting a NAT Gateway	This API is used to delete a NAT gateway.
SNAT Rules	Creating an SNAT Rule	This API is used to create an SNAT rule.
	Querying SNAT Rules	This API is used to query SNAT rules.
	Querying Details About a Specified SNAT Rule	This API is used to query details about an SNAT rule.
	Updating an SNAT Rule	This API is used to update an SNAT rule.
	Deleting an SNAT Rule	This API is used to delete an SNAT rule.
DNAT Rules	Creating a DNAT Rule	This API is used to create a DNAT rule.
	Creating DNAT Rules in Batches	This API is used to create DNAT rules in batches.
	Querying DNAT Rules	This API is used to query DNAT rules.
	Querying Details About a Specified DNAT Rule	This API is used to query details about a DNAT rule.
	Updating a DNAT Rule	This API is used to update a DNAT rule.
	Deleting a DNAT Rule	This API is used to delete a DNAT rule.

API v2.0

Table 2-3 NAT gateway API v2.0

Type	Description
NAT Gateway Service	Create, query, update, and delete NAT gateways, including creating a NAT gateway, querying NAT gateways, querying details about a NAT gateway, updating a NAT gateway, and deleting a NAT gateway.

Type	Description
SNAT Rules	Create, query, and delete SNAT rules, including creating an SNAT rule, querying SNAT rules, querying details about an SNAT rule, and deleting an SNAT rule.
DNAT Rules	Create, query, and delete DNAT rules, including creating a DNAT rule, querying DNAT rules, querying details about a DNAT rule, and deleting a DNAT rule.

Table 2-4 NAT gateway API (v2.0) description

Type	API	Description
NAT Gateway Service	Creating a NAT Gateway	This API is used to create a NAT gateway.
	Querying NAT Gateways	This API is used to query NAT gateways.
	Querying Details About a Specified NAT Gateway	This API is used to query details about a NAT gateway.
	Updating a NAT Gateway	This API is used to update a NAT gateway.
	Deleting a NAT Gateway	This API is used to delete a NAT gateway.
SNAT Rules	Creating an SNAT Rule	This API is used to create an SNAT rule.
	Querying SNAT Rules	This API is used to query SNAT rules.
	Querying Details About a Specified SNAT Rule	This API is used to query details about an SNAT rule.
	Deleting an SNAT Rule	This API is used to delete an SNAT rule.
DNAT Rules	Creating a DNAT Rule	This API is used to create a DNAT rule.
	Querying DNAT Rules	This API is used to query DNAT rules.
	Querying Details About a Specified DNAT Rule	This API is used to query details about a DNAT rule.
	Deleting a DNAT Rule	This API is used to delete a DNAT rule.

Type	API	Description
Tags	Querying NAT Gateways by Tag	This API is used to filter NAT gateways by tag.
	Adding or Deleting NAT Gateway Tags in Batches	This API is used to add tags to or delete tags of a NAT gateway.
	Adding a NAT Gateway Tag	This API is used to add tags to a NAT gateway.
	Deleting a NAT Gateway Tag	This API is used to delete NAT gateway tags.
	Querying NAT Gateway Tags	This API is used to query tags of a NAT gateway.
	Querying Tags in a Project	This API is used to query all tags of a resource type in a region.

3 Calling APIs

3.1 Making an API Request

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

Request URI

A request URI is in the following format:

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

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

Table 3-1 URI parameter description

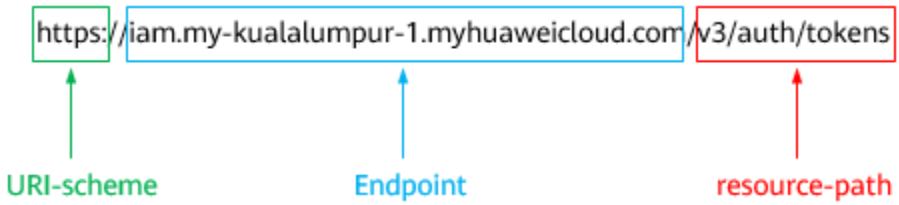
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of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

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

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

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

Figure 3-1 Example URI



NOTE

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

Request Methods

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

Table 3-2 HTTP methods

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

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

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

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

Request Header

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

Common request header fields are as follows.

Table 3-3 Common request header fields

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

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

NOTE

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

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

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

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

(Optional) Request Body

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

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

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

NOTE

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

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

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

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

3.2 Authentication

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

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

Token Authentication

NOTE

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

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

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

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",   // IAM user name
                    "password": "********",  // IAM user password
                    "domain": {
                        "name": "domainname"  // Name of the account to which the IAM user belongs
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxx"     // Project Name
            }
        }
    }
}
```

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

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

AK/SK Authentication

NOTE

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

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

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

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

(Optional) Response Body

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

The following is part of the response body for the API used to **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 V2 APIs of Public NAT Gateways

4.1 NAT Gateway Service

4.1.1 Creating a NAT Gateway

Function

This API is used to create a NAT gateway.

URI

POST /v2/{project_id}/nat_gateways

Table 4-1 Parameter description

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

Request

Table 4-2 describes the request parameters.

Table 4-2 Request parameter

Parameter	Mandatory	Type	Description
nat_gateway	Yes	Object	Specifies the NAT gateway object. For details, see Table 4-3.

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

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	Yes	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	Yes	String	Specifies the VPC ID.
internal_network_id	Yes	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
enterprise_project_id	No	String	• Specifies the enterprise project ID. When creating a NAT gateway, associate an enterprise project ID with the NAT gateway. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Response

Table 4-4 lists response parameters.

Table 4-4 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see **Table 4-5**.

Table 4-5 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the status of the NAT gateway. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.

Parameter	Type	Description
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
enterprise_project_id	String	• Specifies the enterprise project ID. When creating a NAT gateway, associate an enterprise project ID with the NAT gateway. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Examples

- Example request

```
POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/nat_gateways
{
      "nat_gateway": {
      "name": "nat_001",
      "description": "my nat gateway 01",
      "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
      "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
      "spec": "1",
      "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
   }
}
```

- Example response

```
{
   "nat_gateway": {
      "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
      "status": "PENDING_CREATE",
      "description": "my nat gateway 01",
      "admin_state_up": true,
      "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
      "created_at": "2017-11-18 07:34:32.203044",
      "spec": "1",
      "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
      "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "name": "nat_001",
      "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
   }
}
```

Status Codes

See **Status Codes**.

4.1.2 Querying NAT Gateways

Function

This API is used to query NAT gateways. Unless otherwise specified, exact match is applied.

URI

GET /v2/{project_id}/nat_gateways

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 4-6 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
limit	No	Integer	Specifies the number of records on each page.
id	No	String	Specifies the NAT gateway ID.
tenant_id	No	String	Specifies the project ID.
name	No	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	No	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	No	String	Specifies the VPC ID.
internal_network_id	No	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.

Parameter	Mandatory	Type	Description
status	No	String	• Specifies the status of the NAT gateway. • For details about all its values, see Table 7-2.
admin_state_up	No	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	No	String	• Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
enterprise_project_id	No	String	• Specifies the enterprise project ID. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Request

None

Response

Table 4-7 lists response parameters.

Table 4-7 Response parameter

Parameter	Type	Description
nat_gateways	Array(Object)	Specifies the NAT gateway objects. For details, see Table 4-8.

Table 4-8 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.

Parameter	Type	Description
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the status of the NAT gateway. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
enterprise_project_id	String	• Specifies the enterprise project ID. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Examples

- Example request
  GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways?status=ACTIVE

- Example response

```
{
   "nat_gateways": [
      {
         "router_id": "b1d81744-5165-48b8-916e-e56626feb88f",
         "status": "ACTIVE",
         "description": "",
         "admin_state_up": true,
         "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
         "created_at": "2017-11-15 14:50:39.505112",
         "spec": "2",
         "internal_network_id": "5930796a-6026-4d8b-8790-6c6bfc9f87e8",
         "id": "a253be25-ae7c-4013-978b-3c0785eccd63",
         "name": "wj3",
         "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"

      },
      {
         "router_id": "305dc52f-13dd-429b-a2d4-444a1039ba0b",
         "status": "ACTIVE",
         "description": "",
         "admin_state_up": true,
         "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
         "created_at": "2017-11-17 07:41:07.538062",
         "spec": "2",
         "internal_network_id": "fc09463b-4ef8-4c7a-93c8-92d9ca6daf9d",
         "id": "e824f1b4-4290-4ebc-8322-cfff370dbd1e",
         "name": "lyl001",
         "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
      }
   ]
}
```

Status Codes

See **Status Codes**.

4.1.3 Querying Details About a Specified NAT Gateway

Function

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

URI

GET /v2/{project_id}/nat_gateways/{nat_gateway_id}

Table 4-9 Parameter description

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

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.

Request

None

Response

Table 4-10 lists response parameters.

Table 4-10 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see Table 4-11.

Table 4-11 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.

Parameter	Type	Description
spec	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the status of the NAT gateway. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
enterprise_project_id	String	• Specifies the enterprise project ID. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Examples

- Example request
  ```
  GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
  ```
- Example response
  ```
  {
      "nat_gateway": {
          "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
          "status": "ACTIVE",
          "description": "my nat gateway 01",
          "admin_state_up": true,
  ```



```
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:34:32.203044",
        "spec": "1",
        "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
        "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "name": "nat_001",
        "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"

    }
}
```


Status Codes

See **Status Codes**.

4.1.4 Updating a NAT Gateway

Function

This API is used to update a NAT gateway.

NOTE

admin_state_up = True & status = "ACTIVE" can be updated. The name, description, and type of a NAT gateway can be updated.

URI

PUT /v2/{project_id}/nat_gateways/{nat_gateway_id}

Table 4-12 Parameter description

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

Request

Table 4-13 describes the request parameters.

Table 4-13 Request parameter

Parameter	Mandatory	Type	Description
nat_gateway	Yes	Object	Specifies the NAT gateway object. For details, see **Table 4-14**. Mandatory field: None. Only the **name**, **description**, and **spec** fields can be updated. At least one attribute must be specified for the NAT gateway to be updated.

Table 4-14 Description of the **nat_gateway** field

Parameter	Mandatory	Type	Description
name	No	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	No	String	Specifies the NAT gateway type. The value can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.

Response

Table 4-15 lists response parameters.

Table 4-15 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see **Table 4-16**.

Table 4-16 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The value can be: • **1**: small scale • **2**: medium scale • **3**: large scale • **4**: extra-large scale
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the status of the NAT gateway. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Parameter	Type	Description
enterprise_project_id	String	• Specifies the enterprise project ID. The value **0** indicates the default enterprise project. • The value can contain a maximum of 36 characters. It is string **"0"** or in UUID format with hyphens (-).

Examples

- Example request

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways/
a78fb3eb-1654-4710-8742-3fc49d5f04f8
  {
    "nat_gateway": {
        "name": "new_name",
        "description": "new description",
        "spec": "1"
    }
}
```

- Example response

```
{
    "nat_gateway": {
        "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
        "status": "ACTIVE",
        "description": "new description",
        "admin_state_up": true,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:34:32.203044",
        "spec": "1",
        "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
        "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "name": "new_name",
        "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
"
    }
}
```

Status Codes

See **Status Codes**.

4.1.5 Deleting a NAT Gateway

Function

This API is used to delete a NAT gateway.

URI

DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}

Table 4-17 Parameter description

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

Request

None

Response

None

Examples

- Example request
  DELETE https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
- Example response
  None (STATUS CODE 204)

Status Codes

See **Status Codes**.

4.2 SNAT Rules

4.2.1 Creating an SNAT Rule

Function

This API is used to create an SNAT rule.

NOTE

You can create an SNAT rule only when **status** of the NAT gateway is set to **ACTIVE** and **admin_state_up** of the NAT gateway administrator to **True**.

URI

POST /v2/{project_id}/snat_rules

Table 4-18 Parameter description

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

Request

Table 4-19 describes the request parameters.

Table 4-19 Request parameter

Parameter	Mandatory	Type	Description
snat_rule	Yes	Object	Specifies the SNAT rule object. For details, see Table 4-20.

Table 4-20 Description of the **snat_rule** field

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID. For details, see Querying NAT Gateways.
network_id	No	String	Specifies the network ID used by the SNAT rule. Configure either **network_id** or **cidr**.
cidr	No	String	Specifies a CIDR block or a host IP address. This parameter and **network_id** are alternative. If the value of **source_type** is **0**, the CIDR block must be a subset of the VPC subnet CIDR block. If the value of **Source_type** is **1**, **cidr** must be a CIDR block of Direct Connect connection.
source_type	No	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, the default value **0** (VPC) is used.

Parameter	Mandatory	Type	Description
floating_ip_id	Yes	String	Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). The maximum length of the ID is 4096 bytes. The number of EIP IDs cannot exceed 20.
description	No	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.

Response

Table 4-21 lists response parameters.

Table 4-21 Response parameter

Parameter	Type	Description
snat_rule	Object	Specifies the SNAT rule object. For details, see Table 4-22.

Table 4-22 Description of the **snat_rule** field

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, the default value **0** (VPC) is used.

Parameter	Type	Description
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes.
description	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Resource Status Description**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.

Examples

- Example request
  a. Configure parameter **network_id** in a VPC.

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules
{
    "snat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "source_type":0,
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "description": "my snat rule 01"
    }
}
```

  b. Configure parameter **cidr** in a VPC.

```
POST /v2/d199ba7e0ba64899b2e81518104b1526/snat_rules
{
   "snat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "cidr": "192.168.1.10/32",
        "source_type":0,
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "description": "my snat rule 01"
    }
}
```


c. Configure parameter **cidr** over a Direct Connect connection.

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules
 {
     "snat_rule": {
       "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
       "cidr": "172.30.0.0/24",
       "source_type":1,
       "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
       "description": "my snat rule 01"
     }
 }
```

- Example response

a. Response to the request for specifying the **network_id** for a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "description": "",
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
    }
}
```

b. Response to the request for specifying the CIDR block in a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "cidr": "192.168.1.10/32",
        "description": "",
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
    }
}
```

c. Response to the request for specifying the CIDR block in a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "cidr": "172.30.0.0/24",
        "description": "",
        "source_type":1,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
    }
}
```

Status Codes

See **Status Codes**.

4.2.2 Querying SNAT Rules

Function

This API is used to query an SNAT rule list.

URI

GET /v2/{project_id}/snat_rules

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 4-23 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
limit	No	Integer	Specifies the number of records displayed on each page.
id	No	String	Specifies the SNAT rule ID.
tenant_id	No	String	Specifies the project ID.
nat_gateway_id	No	String	Specifies the NAT gateway ID.
network_id	No	String	Specifies the network ID used by the SNAT rule.
cidr	No	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	No	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection.
floating_ip_id	No	String	• Specifies the EIP ID. • Maximum length: 4,096 characters
floating_ip_address	No	String	• Specifies the EIP. • Maximum length: 1,024 characters
status	No	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Table 7-2**.

Parameter	Mandatory	Type	Description
admin_state_up	No	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	No	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Request

None

Response

Table 4-24 lists response parameters.

Table 4-24 Response parameter

Parameter	Type	Description
snat_rules	Array(Object)	Specifies the SNAT rule objects. For details, see **Table 4-25**.

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

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.

Parameter	Type	Description
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, the default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes.
description	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Resource Status Description**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
freezed_ip_address	String	• Specifies the frozen EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules?limit=10
  ```
- Example response
  ```
  {
      "snat_rules": [
  ```



```
    {
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "network_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "source_type":0,
        "tenant_id": "d199ba7e0ba64899b2e81518104b1526",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.242",
        "freezed_ip_address": "",
        "description": "my snat rule 01"
      },
      {
        "floating_ip_id": "6e496fba-abe9-4f5e-9406-2ad8c809ac8c",
        "status": "ACTIVE",
        "nat_gateway_id": "e824f1b4-4290-4ebc-8322-cfff370dbd1e",
        "admin_state_up": true,
        "network_id": "97e89905-f9c8-4ae3-9856-392b0b2fbe7f",
        "source_type":0,
        "tenant_id": "d199ba7e0ba64899b2e81518104b1526",
        "created_at": "2017-11-17 07:43:44.830845",
        "id": "4a1a10d7-0d9f-4846-8cda-24cffeffef5c",
        "floating_ip_address": "5.21.11.142,5.21.11.143",
        "freezed_ip_address": "5.21.11.142",
        "description": "my snat rule 01"
      }
    ]
}
```


Status Codes

See **Status Codes**.

4.2.3 Querying Details About a Specified SNAT Rule

Function

This API is used to query details about a specified SNAT rule.

URI

GET /v2/{project_id}/snat_rules/{snat_rule_id}

Table 4-26 Parameter description

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

Request

None

Response

Table 4-27 lists response parameters.

Table 4-27 Response parameter

Parameter	Type	Description
snat_rule	Object	Specifies the SNAT rule object. For details, see Table 4-28.

Table 4-28 Description of the **snat_rule** field

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, the default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes.
description	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.

Parameter	Type	Description
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Resource Status Description**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
freezed_ip_address	String	• Specifies the frozen EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.

Examples

- Example request

```
GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
```

- Example response

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "source_type":0,
        "tenant_id": "d199ba7e0ba64899b2e81518104b1526",
```



```
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226",
        "freezed_ip_address": "",
        "description": "my snat rule 01"
    }
}
```

Status Codes

See **Status Codes**.

4.2.4 Updating an SNAT Rule

Function

This API is used to update an SNAT rule.

NOTE

You can update the EIP or description only when **status** of the SNAT rule is set to **ACTIVE** and **admin_state_up** of the gateway administrator to **True**.

URI

PUT /v2/{project_id}/snat_rules/{snat_rule_id}

Table 4-29 Parameter description

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

Request

Table 4-30 describes the request parameters.

Table 4-30 Request parameter

Parameter	Mandatory	Type	Description
snat_rule	Yes	Object	Specifies the SNAT rule object. For details, see **Table 4-31**.

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

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
public_ip_address	No	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes. • The number of EIPs cannot exceed 20.
description	No	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.

Response

Table 4-32 lists response parameters.

Table 4-32 Response parameter

Parameter	Type	Description
snat_rule	**NatGatewaySnatRuleResponseBody** object	Specifies the response body of the SNAT rule.

Table 4-33 NatGatewaySnatRuleResponseBody

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.

Parameter	Type	Description
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, the default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes.
description	String	Provides supplementary information about the SNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
public_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.

Examples

- Example request

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
{
    "snat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "description": "my snat rule 01",
        "public_ip_address": "10.15.10.11,10.15.10.12"
    }
}
```

- Example response

```
{
    "snat_rule": {
        "floating_ip_id": " bdc10a4c-d81a-41ec-adf7-de857f7c812a,7a094014-9657-463f-972b-e84d56b931a0",
        "status": "PENDING_UPDATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "public_ip_address": "10.15.10.11,10.15.10.12",        ,
        "floating_ip_address": "10.15.10.11,10.15.10.12",
        "description": "my snat rule 01"
    }
}
```

Status Codes

See **Status Codes**.

4.2.5 Deleting an SNAT Rule

Function

This API is used to delete an SNAT rule.

URI

DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}/snat_rules/{snat_rule_id}

Table 4-34 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
nat_gateway_id	Yes	String	Specifies the ID of the NAT gateway to which the SNAT rule belongs.
snat_rule_id	Yes	String	Specifies the SNAT rule ID.

Request

None

Response

None

Examples

- Example request
  DELETE https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways/f4dfea98-874a-46f7-aa2a-fb348d0ceb02/snat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
- Example response
  None (STATUS CODE 204)

Status Codes

See **Status Codes**.

4.3 DNAT Rules

4.3.1 Creating a DNAT Rule

Function

This API is used to create a DNAT rule.

NOTE

You can create a DNAT rule only when **status** of the NAT gateway is set to **ACTIVE** and **admin_state_up** of the NAT gateway administrator to **True**. Specify either **port_id** or **private_ip** at a time. If you are going to create a DNAT rule that allows traffic to and from all ports of a server and an EIP, set **internal_service_port** to **0**, **external_service_port** to **0**, and **protocol** to **any**.

URI

POST /v2/{project_id}/dnat_rules

Table 4-35 Parameter description

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

Request

Table 4-36 lists the request parameter.

Table 4-36 Request parameter

Parameter	Mandatory	Type	Description
dnat_rule	Yes	Object	Specifies the DNAT rule object. For details, see **Table 4-37**.

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

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
port_id	No	String	Specifies the port ID of an ECS or a BMS. This parameter and **private_ip** are alternative.
private_ip	No	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection. You can specify either this parameter or **port_id**.
internal_service_port	Yes	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems. The value ranges from **0** to **65535**.
floating_ip_id	Yes	String	Specifies the EIP ID.
external_service_port	Yes	Integer	Specifies the port for providing services for external systems. The value ranges from **0** to **65535**.
protocol	Yes	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	No	String	Provides supplementary information about the DNAT rule.

Parameter	Mandatory	Type	Description
internal_service_port_range	No	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The number of ports must be the same as that of **external _service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.
external_service_port_range	No	String	Specifies the port range used by the floating IP address for providing services for external systems. • The number of ports must be same as that of **internal _service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.

Response

Table 4-38 lists response parameter.

Table 4-38 Response parameter

Parameter	Type	Description
dnat_rule	Object	Specifies the DNAT rule object. For details, see **Table 4-39**.

Table 4-39 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.

Parameter	Type	Description
port_id	String	Specifies the port ID of an ECS or a BMS. This parameter and **private_ip** are alternative.
private_ip	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP address.
external_service_port	Integer	Specifies the port for providing services for external systems.
protocol	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	String	Provides supplementary information about the DNAT rule.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Parameter	Type	Description
internal_service_port_range	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. ● The number of ports must be the same as that of **external_service_port_range**. ● The value ranges from **1** to **65535**.
external_service_port_range	String	Specifies the port range used by the floating IP address for providing services for external systems. ● The number of ports must be same as that of **internal_service_port_range**. ● The value ranges from **1** to **65535**.

Examples

- Example requests
  a. Creating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules
{
    "dnat_rule": {
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    }
}
```

  b. Creating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules
{
    "dnat_rule": {
        "floating_ip_id": "Cf99c679-9f41-4dac-8513-9c9228e713e1",
        "nat_gateway_id": "Dda3a125-2406-456c-a11f-598e10578541",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "external_service_port": 0,
        "description": "my dnat rule 01"
    }
}
```

  c. Creating a DNAT rule with specified **external_service_port_range** and **internal_service_port_range**

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules
{
    "dnat_rule": {
```



```
        "floating_ip_id": "Cf99c679-9f41-4dac-8513-9c9228e713e1",
        "nat_gateway_id": "Dda3a125-2406-456c-a11f-598e10578541",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "tcp",
        "external_service_port": 0,
        "description": "my dnat rule 01" ,
        "external_service_port_range":"100-200",
        "internal_service_port_range":"100-200"
    }
}
```

- Example responses

  a. Response to the request for creating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
{
    "dnat_rule": {
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "external_service_port": 242,
        "floating_ip_address": "5.21.11.226",
        "description": "my dnat rule 01"
    }
}
```

  b. Response to the request for creating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**

```
{
    "dnat_rule": {
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "external_service_port": 0,
        "floating_ip_address": "5.21.11.227",
        "description": "my dnat rule 01"
    }
}
```

  c. Response to the request for creating a DNAT rule with specified **external_service_port_range** and **internal_service_port_range**

```
{
    "dnat_rule": {
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "tcp",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "external_service_port": 0,
        "floating_ip_address": "5.21.11.227",
```



```
        "description": "my dnat rule 01",
        "internal_service_port_range":"100-200",
        "external_service_port_range":"100-200"
    }
}
```

Status Codes

See **Status Codes**.

4.3.2 Creating DNAT Rules in Batches

Function

This API is used to create DNAT rules in batches.

NOTE

You can create DNAT rules in batches only when **status** of the NAT gateway is set to **ACTIVE** and **admin_state_up** of the NAT gateway administrator to **True**. Specify either **port_id** or **private_ip** at a time. If you are going to create a DNAT rule that allows traffic to and from all ports of a server and an EIP, set **internal_service_port** to **0**, **external_service_port** to **0**, and **protocol** to **any**.

URI

POST /v2/{project_id}/dnat_rules/batch

Table 4-40 Parameter description

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

Request

Table 4-41 lists the request parameter.

Table 4-41 Request parameter

Parameter	Mandatory	Type	Description
dnat_rules	Yes	Array of **dnat_rule** objects	Specifies objects of DNAT rules. For details, see **Table 4-42**.

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

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
port_id	No	String	Specifies the port ID of an ECS or a BMS. This parameter and **private_ip** are alternative.
private_ip	No	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection. You can specify either this parameter or **port_id**.
internal_service_port	Yes	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	Yes	String	Specifies the EIP ID.
external_service_port	Yes	Integer	Specifies the port for providing services for external systems.
protocol	Yes	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	No	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
internal_service_port_range	No	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The number of ports must be the same as that of **external _service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.

Parameter	Mandatory	Type	Description
external_service_port_range	No	String	Specifies the port range used by the floating IP address for providing services for external systems. • The number of ports must be same as that of **internal _service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.

Response

Table 4-43 lists response parameter.

Table 4-43 Response parameter

Parameter	Type	Description
dnat_rules	Array(dnat_rule)	Specifies objects of DNAT rules. For details, see **Table 4-44**.

Table 4-44 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP address.

Parameter	Type	Description
external_service_port	Integer	Specifies the port for providing services for external systems.
protocol	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
internal_service_port_range	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The number of ports must be the same as that of **external _service_port_range**. • The value ranges from **1** to **65535**.
external_service_port_range	String	Specifies the port range used by the floating IP address for providing services for external systems. • The number of ports must be same as that of **internal _service_port_range**. • The value ranges from **1** to **65535**.

Examples

- Example requests
  a. Creating rules in batches (In the first DNAT rule, both **internal_service_port** and **external_service_port** are set to a specific port number. In the second DNAT rule, both **internal_service_port** and **external_service_port** are set to **0**.)

```
POST https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules/batch
{
    "dnat_rules": [{
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "external_service_port": 242
```



```
    },
    {
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "external_service_port": 0
    }]
}
```

- Example responses

  a. Response to the request for creating DNAT rules in batches

```
{
    "dnat_rules": [{
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "private_ip": "",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    },
{
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "port_ID": "",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.227",
        "external_service_port": 0,
        "description": "my dnat rule 01"
    }]
}
```

Status Codes

See **Status Codes**.

4.3.3 Querying DNAT Rules

Function

This API is used to query DNAT rules.

URI

GET /v2/{project_id}/dnat_rules

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 4-45 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
limit	No	Integer	Specifies the number of records displayed on each page.
id	No	String	Specifies the DNAT rule ID.
tenant_id	No	String	Specifies the project ID.
nat_gateway_id	No	String	Specifies the NAT gateway ID.
port_id	No	String	Specifies the port ID of an ECS or a BMS.
private_ip	No	String	Specifies the private IP address of a user, for example, the IP address of a VPC for Direct Connect connection.
internal_service_port	No	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	No	String	Specifies the EIP ID.
floating_ip_address	No	String	Specifies the EIP.
external_service_port	No	Integer	Specifies the port for providing external services.
protocol	No	String	Specifies the protocol type. TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
description	No	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
status	No	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Resource Status Description**.

Parameter	Mandatory	Type	Description
admin_state_up	No	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.

Request

None

Response

Table 4-46 lists response parameters.

Table 4-46 Response parameter

Parameter	Type	Description
dnat_rules	Array(Object)	Specifies the DNAT rule objects. For details, see Table 4-47.

Table 4-47 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address of a user, for example, the IP address of a VPC for Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.

Parameter	Type	Description
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP.
external_service_port	Integer	Specifies the port for providing external services.
protocol	String	Specifies the protocol type. TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
description	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Parameter	Type	Description
internal_service_port_range	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. ● The port range is the same as the value of **external_service_port_range**. ● The value ranges from 1 to 65535.
external_service_port_range	String	Specifies the port range used by the floating IP address for providing external services. ● The port range is the same as the value of **internal_service_port_range**. ● The value ranges from 1 to 65535.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/dnat_rules?limit=10
  ```
- Example response
  ```
  {
      "dnat_rules": [
          {
              "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
              "status": "ACTIVE",
              "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
              "admin_state_up": true,
              "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
              "private_ip": "",
              "internal_service_port": 993,
              "protocol": "tcp",
              "tenant_id": "abc",
              "created_at": "2017-11-15 15:44:42.595173",
              "id": "79195d50-0271-41f1-bded-4c089b2502ff",
              "floating_ip_address": "5.21.11.226",
              "external_service_port": 242,
              "description": "my dnat rule 01"
          },
          {
              "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
              "status": "ACTIVE",
              "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
              "admin_state_up": true,
              "port_id": "",
              "private_ip": "192.168.1.100",
              "internal_service_port": 0,
              "protocol": "any",
              "tenant_id": "abc",
  ```



```
        "created_at": "2017-11-16 15:44:42.595173",
        "id": "89195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.227",
        "external_service_port": 0,
        "description": "my dnat rule 01"
    }
  ]
}
```

Status Codes

See **Status Codes**.

4.3.4 Querying Details About a Specified DNAT Rule

Function

This API is used to query details about a specified DNAT rule.

URI

GET /v2/{project_id}/dnat_rules/{dnat_rule_id}

Table 4-48 Parameter description

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

Request

None

Response

Table 4-49 lists response parameters.

Table 4-49 Response parameter

Parameter	Type	Description
dnat_rule	Object	Specifies the DNAT rule object. For details, see **Table 4-50**.

Table 4-50 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address of a user, for example, the IP address of a VPC for Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP.
external_service_port	Integer	Specifies the port for providing external services.
protocol	String	Specifies the protocol type. TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
description	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.

Parameter	Type	Description
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
internal_service_port_range	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The port range is the same as the value of **external_service_port_range**. • The value ranges from 1 to 65535.
external_service_port_range	String	Specifies the port range used by the floating IP address for providing external services. • The port range is the same as the value of **internal_service_port_range**. • The value ranges from 1 to 65535.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/dnat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
  ```
- Example response
  ```
  {
      "dnat_rules": {
          "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
          "status": "ACTIVE",
          "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
          "admin_state_up": true,
  ```



```
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "private_ip": "",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "d199ba7e0ba64899b2e81518104b1526d",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    }
}
```


Status Codes

See **Status Codes**.

4.3.5 Updating a DNAT Rule

Function

This API is used to update a DNAT rule.

NOTE

You can update a DNAT rule only when its **status** is set to **ACTIVE** and **admin_state_up** of the NAT gateway administrator to **True**. Specify either **port_id** or **private_ip** at a time. If you are going to create a DNAT rule that allows traffic to and from all ports of a server and an EIP, set **internal_service_port** to **0**, **external_service_port** to **0**, and **protocol** to **any**.

The fields, including **port_id**, **private_ip**, **internal_service_port**, **external_service_port**, **floating_ip_id**, **protocol**, **internal_service_port_range**, and **external_service_port_range** must be updated together.

URI

PUT /v2/{project_id}/dnat_rules/{dnat_rule_id}

Table 4-51 Parameter description

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

Request

Table 4-52 lists the request parameter.

Table 4-52 Parameter description

Parameter	Mandatory	Type	Description
dnat_rule	Yes	Object	Specifies the DNAT rule object. For details, see **Table 4-53**.

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

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
port_id	No	String	Specifies the port ID of an ECS or a BMS. This parameter and **private_ip** are alternative.
private_ip	No	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection. You can specify either this parameter or **port_id**.
internal_service_port	No	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	No	String	Specifies the EIP ID.
external_service_port	No	Integer	Specifies the port for providing services for external systems.
protocol	No	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	No	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.

Parameter	Mandatory	Type	Description
internal_service_port_range	No	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The number of ports must be the same as that of **external_service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.
external_service_port_range	No	String	Specifies the port range used by the floating IP address for providing services for external systems. • The number of ports must be same as that of **internal _service_port_range**. • The value ranges from **1** to **65535**. • Specify two port numbers connected by a single hyphen (-) and no blank spaces in the *x-y* format, where *x* is lower than *y*.

Response

Table 4-54 lists response parameter.

Table 4-54 Response parameter

Parameter	Type	Description
dnat_rule	**NatGatewayDnatRuleResponseBody** object	Specifies the response body of the DNAT rule.

Table 4-55 NatGatewayDnatRuleResponseBody

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.

Parameter	Type	Description
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP address.
external_service_port	Integer	Specifies the port for providing services for external systems.
protocol	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).
description	String	Provides supplementary information about the DNAT rule. You can enter up to 255 characters.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies the unfrozen or frozen state. • The value can be: – **true**: indicates the unfrozen state. – **false**: indicates the frozen state.
created_at	String	• Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.
internal_service_port_range	String	Specifies the port range used by ECSs or BMSs to provide services for external systems. • The number of ports must be the same as that of **external _service_port_range**. • The value ranges from **1** to **65535**.
external_service_port_range	String	Specifies the port range used by the floating IP address for providing services for external systems. • The number of ports must be same as that of **internal _service_port_range**. • The value ranges from **1** to **65535**.

Examples

- Example requests
  a. Updating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules/
79195d50-0271-41f1-bded-4c089b2502ff
{
    "dnat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    }
}
```

  b. Updating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules/
79195d50-0271-41f1-bded-4c089b2502ff
{
    "dnat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "external_service_port": 0,
       "description": "my dnat rule 01"
    }
}
```

  c. Updating a DNAT rule with specified **internal_service_port_range** and **external_service_port_range**

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules/
79195d50-0271-41f1-bded-4c089b2502ff
{
    "dnat_rule": {
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "tcp",
        "external_service_port": 0,
        "description": "my dnat rule 01" ,
        "external_service_port_range":"100-200",
        "internal_service_port_range":"100-200"
    }
}
```

  d. Updating the description of a DNAT rule

```
PUT https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/dnat_rules/
79195d50-0271-41f1-bded-4c089b2502ff
{
    "dnat_rule": {
       "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
       "description": "my dnat rule 01"
    }
}
```

- Example responses
  a. Response to the request for updating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
{
    "dnat_rule": {
```



```
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "abc",
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    }
}
```

b. Response to the request for updating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**

```
{
    "dnat_rule": {
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "tenant_id": "abc",
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address ": "5.21.11.227",
        "external_service_port": 0,
        "description": "my dnat rule 01"
    }
}
```

c. Response to the request for updating a DNAT rule with specified **external_service_port_range** and **internal_service_port_range**

```
{
    "dnat_rule": {
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "tcp",
        "tenant_id": "abc",
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address ": "5.21.11.227",
        "external_service_port": 0,
        "description": "my dnat rule 01",
        "internal_service_port_range": "100-200",
        "external_service_port_range": "100-200"
    }
}
```

d. Response to request for updating the description of a DNAT rule

```
{
    "dnat_rule": {
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "private_ip": "",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "abc",
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
```



```
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242,
        "description": "my dnat rule 01"
    }
}
```

Status Codes

See **Status Codes**.

4.3.6 Deleting a DNAT Rule

Function

This API is used to delete a DNAT rule.

URI

DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}/dnat_rules/{dnat_rule_id}

Table 4-56 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
nat_gateway_id	Yes	String	Specifies the ID of the NAT gateway to which the DNAT rule belongs.
dnat_rule_id	Yes	String	Specifies the DNAT rule ID.

Request

None

Response

None

Examples

- Example request
  ```
  DELETE https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/nat_gateways/f4dfea98-874a-46f7-aa2a-fb348d0ceb02/dnat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
  ```
- Example response
  ```
  None (STATUS CODE 204)
  ```


Status Codes

See **Status Codes**.

5 Application Examples

5.1 Creating an SNAT Rule Using a Specific Subnet

Scenarios

This section describes how to create an SNAT rule by calling APIs. For details about how to call APIs, see **Calling APIs**.

Prerequisites

- You have created a VPC and a subnet. For details, see *Virtual Private Cloud API Reference*.
- Ensure that the VPC does not have a default route.
- You can create a rule only when **status** of the NAT gateway is **ACTIVE** and **admin_state_up** of the NAT gateway administrator is **True**.
- If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header when making an API call. Obtain the token by referring to **Authentication**.

NOTE

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

Procedure

Step 1 Create a NAT gateway.

1. Determine the VPC to be used.
   - Query VPCs.

     URI format: GET /v1/{project_id}/vpcs

     For details, see "Querying VPCs" in the *Virtual Private Cloud API Reference*.
   - Select a VPC that does not have a default route, and make a note of the VPC ID.

2. Determine the subnet in the VPC to be used.
   - Query the subnets in the VPC.

     URI format: GET /v1/{project_id}/subnets?vpc_id={vpc_id}

     For details, see "Querying Subnets" in the *Virtual Private Cloud API Reference*.
   - Select a subnet based on service requirements and make a note of the subnet ID.
3. Create a NAT gateway.
   - API

     URI format: POST /v2/{project_id}/nat_gateways

     For details, see **Creating a NAT Gateway**.
   - Example request

     POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/nat_gateways

     Obtain *{endpoint}* from **Regions and Endpoints**.
   - Example response

```
{
   "nat_gateway": {
      "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
      "status": "PENDING_CREATE",
      "description": "my nat gateway 01",
      "admin_state_up": true,
      "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
      "created_at": "2017-11-18 07:34:32.203044",
      "spec": "1",
      "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
      "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "name": "nat_001",
      "dnat_rules_limit": "200",
      "snat_rule_public_ip_limit": "20",
      "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2",
      "billing_info": ""
   }
}
```

4. Ensure that the NAT gateway is successfully created and is in the active state.

   API

   URI format: GET /v2/{project_id}/nat_gateways/{nat_gateway_id}

   For details, see **Querying Details About a Specified NAT Gateway**.
5. Make a note of the NAT gateway ID and the value of **internal_network_id**.

Step 2 Determine the EIP to be used.

1. Query EIPs.

   API

   URI format: GET /v1/{project_id}/publicips

   For details, see "Querying EIPs" in the *Elastic IP API Reference*.
2. Select an EIP based on service requirements and make a note of the EIP ID.

Step 3 Create an SNAT rule.

- API

  URI format: POST /v2/{project_id}/snat_rules

For details, see **Creating an SNAT Rule**.

- Example request

  POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/snat_rules

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:

```
{
   "snat_rule": {
      "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
      "source_type": 0,
      "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
      "description": "my snat rule 01"
   }
}
```

- Example response

```
{
   "snat_rule": {
      "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
      "status": "PENDING_CREATE",
      "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "admin_state_up": true,
      "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
      "description": "",
      "source_type": 0,
      "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
      "created_at": "2017-11-18 07:54:21.665430",
      "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
      "floating_ip_address": "5.21.11.226"
   }
}
```

Step 4 Verify that the SNAT rule is successfully created.

- API

  URI format: GET /v2/{project_id}/snat_rules/{snat_rule_id}

  For details, see .

- Example request

  GET https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338

  Obtain *{endpoint}* from **Regions and Endpoints**.

- Example response

```
{
   "snat_rule": {
      "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
      "status": "ACTIVE",
      "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "admin_state_up": true,
      "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
      "source_type": 0,
      "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
      "created_at": "2017-11-18 07:54:21.665430",
      "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
      "floating_ip_address": "5.21.11.226",
      "freezed_ip_address": "",
      "description": "my snat rule 01"
   }
}
```

----End

5.2 Creating a DNAT Rule Using a Specific ECS

Scenarios

This section describes how to create a DNAT rule by calling APIs. For details about how to call APIs, see **Calling APIs**.

Prerequisites

- You have created a VPC and a subnet. For details, see *Virtual Private Cloud API Reference*.
- Ensure that the VPC does not have a default route.
- You can create a rule only when **status** of the NAT gateway is **ACTIVE** and **admin_state_up** of the NAT gateway administrator is **True**.
- If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header when making an API call. Obtain the token by referring to **Authentication**.

NOTE

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

Procedure

Step 1 Create a NAT gateway.

1. Determine the VPC to be used.
   - Query VPCs.

     URI format: GET /v1/{project_id}/vpcs

     For details, see "Querying VPCs" in the *Virtual Private Cloud API Reference*.
   - Select a VPC that does not have a default route, and make a note of the VPC ID.
2. Determine the subnet in the VPC to be used.
   - Query the subnets in the VPC.

     URI format: GET /v1/{project_id}/subnets?vpc_id={vpc_id}

     For details, see "Querying Subnets" in the *Virtual Private Cloud API Reference*.
   - Select a subnet based on service requirements and make a note of the subnet ID.
3. Create a NAT gateway.
   - API

     URI format: POST /v2/{project_id}/nat_gateways

     For details, see **Creating a NAT Gateway**.
   - Example request

     POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/nat_gateways

Obtain *{endpoint}* from **Regions and Endpoints**.

– Example response

```
{
    "nat_gateway": {
        "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
        "status": "PENDING_CREATE",
        "description": "my nat gateway 01",
        "admin_state_up": true,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:34:32.203044",
        "spec": "1",
        "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
        "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "name": "nat_001",
        "dnat_rules_limit": "200",
        "snat_rule_public_ip_limit": "20",
        "enterprise_project_id": "0aad99bc-f5f6-4f78-8404-c598d76b0ed2",
        "billing_info": ""
    }
}
```

4. Ensure that the NAT gateway is successfully created and is in the active state.

   API

   URI format: GET /v2/{project_id}/nat_gateways/{nat_gateway_id}

   For details, see **Querying Details About a Specified NAT Gateway**.

5. Make a note of the NAT gateway ID and the value of **internal_network_id**.

Step 2 Determine the ECS to be used.

1. Query ECSs.

   API

   URI format: GET https://{endpoint}/v1/{project_id}/cloudservers/detail

   For details, see "Querying Details About ECSs" in the *Elastic Cloud Server API Reference*.

2. Select an ECS based on service requirements and make a note of the port ID of the ECS NIC.

Step 3 Determine the EIP to be used.

1. Query EIPs.

   API

   URI format: GET /v1/{project_id}/publicips

   For details, see "Querying EIPs" in the *Elastic IP API Reference*.

2. Select an EIP based on service requirements and make a note of the EIP ID.

Step 4 Create a DNAT rule.

- API

  URI format: POST /v2/{project_id}/dnat_rules

  For details, see **Creating a DNAT Rule**.

- Example request

  POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/dnat_rules

  Obtain *{endpoint}* from **Regions and Endpoints**.

  Body:


```
{
   "dnat_rule": {
      "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
      "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
      "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
      "internal_service_port": 993,
      "protocol": "tcp",
      "external_service_port": 242,
      "description": "my dnat rule 01"
   }
}
```

- Example response

```
{
   "dnat_rule": {
      "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
      "status": "PENDING_CREATE",
      "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
      "admin_state_up": true,
      "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
      "internal_service_port": 993,
      "protocol": "tcp",
      "tenant_id": "abc",
      "created_at": "2017-11-15 15:44:42.595173",
      "id": "79195d50-0271-41f1-bded-4c089b2502ff",
      "external_service_port": 242,
      "floating_ip_address": "5.21.11.226",
      "description": "my dnat rule 01"
   }
}
```

Step 5 Verify that the DNAT rule is successfully created.

- API

  URI format: GET /v2/{project_id}/dnat_rules/{dnat_rule_id}

  For details, see **Querying Details About a Specified DNAT Rule**.

- Example request

  GET https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/dnat_rules/79195d50-0271-41f1-bded-4c089b2502ff

  Obtain *{endpoint}* from **Regions and Endpoints**.

- Example response

```
{
   "dnat_rule": {
      "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
      "status": "ACTIVE",
      "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
      "admin_state_up": true,
      "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
      "internal_service_port": 993,
      "protocol": "tcp",
      "tenant_id": "abc",
      "created_at": "2017-11-15 15:44:42.595173",
      "id": "79195d50-0271-41f1-bded-4c089b2502ff",
      "external_service_port": 242,
      "floating_ip_address": "5.21.11.226",
      "description": "my dnat rule 01"
   }
}
```

----End

6 Permissions Policies and Supported Actions

6.1 Introduction

This section describes fine-grained permissions management for your NAT gateways. If your account does not need individual IAM users, then you may skip this section.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all the permissions required to call all APIs, but IAM users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query NAT gateways using an API, the user must have been granted permissions that allow the **nat:natGateways:list** action.

Supported Actions

NAT Gateway provides system-defined policies, which can be directly used in IAM. The accountaccount administrator can also create custom policies to supplement system-defined policies for more refined access control. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: Statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.
- Actions: Specific operations that are allowed or denied.

- IAM or enterprise projects: Type of projects for which an action will take effect. Policies that contain actions for both IAM and enterprise projects can be used and take effect for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM. The accountaccount administrator can check whether an action supports IAM projects or enterprise projects in the action list. The check mark (√) indicates that the action supports the project and the cross symbol (×) indicates that the action does not support the project.

NAT Gateway supports the following actions that can be defined in custom policies:

- **NAT Gateway v2**, including actions supported by all v2 APIs of the NAT gateway, such as creating, updating, and deleting NAT gateways.
- **SNAT Rule v2**, including actions supported by all v2 APIs of the SNAT rule, such as creating and querying SNAT rules.
- **DNAT Rule v2**, including actions supported by all v2 APIs of the DNAT rule, such as creating and querying DNAT rules.

6.2 NAT Gateway v2

Permission	API	Action	IAM Project	Enterprise Project
Creating a NAT gateway	POST /v2/{project_id}/nat_gateways	nat:natGateways:create	√	√
Querying NAT gateways	GET /v2/{project_id}/nat_gateways	nat:natGateways:list	√	√
Querying details about a NAT gateway	GET /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:get	√	√
Updating a NAT gateway	PUT /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:update	√	√

Permission	API	Action	IAM Project	Enterprise Project
Deleting a NAT gateway	DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:delete	√	√

6.3 SNAT Rule v2

Permission	API	Action	IAM Project	Enterprise Project
Creating an SNAT rule	POST /v2/{project_id}/snat_rules	nat:snatRules:create	√	√
Querying SNAT rules	GET /v2/{project_id}/snat_rules	nat:snatRules:list	√	√
Querying details about an SNAT rule	GET /v2/{project_id}/snat_rules/{snat_rule_id }	nat:snatRules:get	√	√
Deleting an SNAT rule	DELETE /v2/{project_id}/snat_rules/{snat_rule_id}	nat:snatRules:delete	√	√
Updating an SNAT rule	PUT /v2/{project_id}/snat_rules/{snat_rule_id}	nat:snatRules:update	√	√

6.4 DNAT Rule v2

Permission	API	Action	IAM Project	Enterprise Project
Creating a DNAT rule	POST /v2/{project_id}/dnat_rules	nat:dnatRules:create	√	√

Permission	API	Action	IAM Project	Enterprise Project
Creating DNAT rules in batches	POST /v2/{project_id}/dnat_rules/batch	nat:dnatRules:create	√	√
Querying DNAT rules	GET /v2/{project_id}/dnat_rules	nat:dnatRules:list	√	√
Querying details about a DNAT rule	GET /v2/{project_id}/dnat_rules/{dnat_rule_id }	nat:dnatRules:get	√	√
Deleting a DNAT rule	DELETE /v2/{project_id}/dnat_rules/{dnat_rule_id}	nat:dnatRules:delete	√	√
Updating a DNAT rule	PUT /v2/{project_id}/dnat_rules/{dnat_rule_id}	nat:dnatRules:update	√	√

7 Common Parameters

7.1 Status Codes

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

Error Response Code	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.
406 Not Acceptable	The response generated by the server cannot be accepted by the client.

Error Response Code	Description
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 cannot 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 system is currently unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

7.2 Error Code Description v2

Background Information

- An error code returned by an API does not correspond to one error message. The following table lists only common error messages.
- Most NAT Gateway APIs are asynchronous. Some error codes are displayed in the returned messages for task viewing requests. HTTP status codes may not be accurate.
- The NAT Gateway service is strongly dependent on other services, such as network and storage. When error messages are provided for the NAT Gateway-depended services, contact technical support for troubleshooting.

Error Codes

Table 7-1 Error code description v2

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
Public	400	NAT.0001	Invalid timestamp.	Invalid value for created_at %(timestamp)s.	Enter the time in the correct format.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0002	Invalid request parameters.	Invalid parameters.	Enter the correct parameter.
	404	NAT.0004	The router does not exist.	The router %(router_id)s does not exist.	Check whether the entered router ID is correct.
	404	NAT.0005	The network does not exist.	Network %(network_id)s does not exist.	Enter a valid network ID.
	400	NAT.0006	Rule has not been deleted.	Rule has not been deleted.	Check whether the gateway fails to be deleted because its rules have not been deleted.
	400	NAT.0007	The database is abnormal.	DB Error	Contact technical support.
	400	NAT.0008	The subnet is not connected to the virtual router.	Router %(router)s has no port for subnet %(subnet)s.	Add the subnet to the router port.
	400	NAT.0009	The resource is in use.	Resource %(res_type)s %(res)s is used by %(user_type)s %(user)s	Check whether the resource is in use.
	400	NAT.0010	The subnet does not exist in the network.	Network %(network)s does not contain any IPv4 subnet	Check whether the subnet has been added to the network.
	400	NAT.0012	A NAT gateway has been created for the network.	The network %(network)s already has nat gateway.	Select a network in which no NAT gateway has been created.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	404	NAT.0013	The NAT gateway cannot be found.	Router % (router)s for the specified NAT gateway could not be found.	The NAT gateway cannot be found.
	400	NAT.0014	The description contains more than 255 characters.	Invalid input for description. +exceeds maximum length of 255.	Enter a maximum of 255 characters.
	400	NAT.0015	The name contains more than 255 characters.	Invalid input for name.+exceeds maximum length of 255.	Enter a maximum of 255 characters.
	400	NAT.0016	The **spec** value is not **1**, **2**, **3**, or **4**.	Invalid input for spec. Reason: '*' is not in ['1', '2', '3', '4'].	Check whether the **spec** value is **1**, **2**, **3**, or **4**.
	400	NAT.0017	Invalid router ID.	Invalid input for router_id. Reason: '********-****-****-****-************' is not a valid UUID.	Enter a correct router ID.
	400	NAT.0018	Invalid internal network ID.	Invalid input for internal_network_id. Reason: '********-****-****-****-************' is not a valid UUID.	Enter a correct internal network ID.
	404	NAT.0019	The network ID does not exist.	Network ********-****-****-****-************ could not be found.	Check whether the internal network ID exists.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	404	NAT.0020	The **tenant_id** value is empty or the parameter does not exist.	Specifying 'tenant_id' other than authenticated tenant in request requires admin privileges	Check whether the tenant ID exists.
	404	NAT.0021	The **nat_gateway_id** value is empty or the parameter does not exist.	Invalid input for nat_gateway_id. Reason: '********-****-****-****-************'is not a valid UUID.	Check whether the NAT gateway ID exists.
	400	NAT.0022	The CIDR block of the SNAT rule conflicts with the network.	Either network_id or cidr must be specified. Both can not be specified at the same time	Enter either the network ID or CIDR block.
	404	NAT.0023	The port ID does not exist.	Port '********-****-****-****-************' could not be found.	Check whether the port ID exists.
	404	NAT.0024	The **floating_ip_id** value is empty or the parameter does not exist or is invalid.	Invalid input for floating_ip_id. Reason: '********-****-****-****-************'is not a valid UUID.	Check whether the floating IP address ID exists.
	401	NAT.0025	The token has expired.	Token is expired.	Check whether the token is within validity period.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0026	The floating IP address ID does not exist.	Floating IP ********-****-****-****-************ could not be found.	Check whether the floating IP address ID exists.

Nat Gateway	400	NAT.0101	NAT gateway request error.	Lack of user authority.
				//request is null.
				//endpoint is empty.
				// resource type is invalid. // create natgateway request is null. // update natgateway request is null
					Troubleshoot the fault as prompted or contact technical support.

	400	NAT.0102	Try again later.	The system is busy. Please try again later.	Try again later.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution

	400	NAT.0101	Incorrect NAT gateway parameter.	Request is invalid. // NatGateway id invalid.
				// the enterprise project id is unsupported.
				// the enterprise project id in request is invalid. // parameter is null.
				// tags is invalid.
				// get natgateways error limit is invalid.
				//get natgateways error marker is invalid.
				//Only admin user can do this action. // Parameters are invalid, check them and try.
					Enter the correct parameter or contact technical support.

	400	NAT.0103	The NAT gateway is not activated.	NatGateway %(nat_gateway_id)s is not ACTIVE.	Check the gateway status. If the gateway is not in the running state for a long time, contact technical support.
	400	NAT.0104	The NAT gateway is not in the UP state.	NatGateway %(nat_gateway_id)s is not UP.	Check the gateway status. The gateway may be frozen due to arrears.
	400	NAT.0104	The NAT gateway is frozen.	NatGateway %(nat_gateway_id)s is frozen.can not update	Check the gateway status. The gateway may be frozen due to arrears and cannot be updated.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400/404	NAT.0105	The NAT gateway does not exist. (The HTTP status code 400 indicates that the gateway does not exist when deleting the gateway. The HTTP status code 404 indicates that the gateway does not exist when querying the gateway.)	NatGateway %(nat_gateway_id)s does not exist.	Check whether the NAT gateway ID exists.
	400	NAT.0106	Concurrent operation conflicts.	Concurrent conflict requests found	Contact technical support.
	400	NAT.0107	Failed to create the internal port of the NAT gateway.	Create NG Port failed.	Internal error. Contact technical support.
	400	NAT.0108	Failed to bind the internal port to the NAT gateway.	NG Port %(port)s is unbound.	Internal error. Contact technical support.
	400	NAT.0109	The NAT gateway does not support IPv6.	NatGateway does not support IPv6.	The NAT gateway cannot be bound to an IPv6 EIP.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0110	An error occurred when selecting the gateway node.	Get Nat gateway host failed	Contact technical support.
	400	NAT.0111	Failed to obtain the IP address of the gateway node.	Get Nat gateway agent local_ip failed	Contact technical support.
	400	NAT.0112	Failed to obtain the VPC route table.	Get RouteTable %(router_id)s failed.	Contact technical support.
	400	NAT.0113	The number of NAT gateways exceeds the upper limit.	%(limit)s NAT gateways has been created to this VPC, no more is allowed	Create a NAT gateway under another VPC or delete the existing NAT gateways under the VPC.

Snat Rule	400	NAT.0201	Incorrect SNAT rule parameter.	Endpoint is null or empty. // Endpoint is Invalid.
				//Request is null. // natGatewayId is invalid. // SnatRule id invalid. // NatGatewayId is invalid.
				//Invalid value for public ip id.
				//Endpoint is null.
				//request is null.
				//Query SnatRules list error marker is invalid.
					Enter the correct parameter or contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0202	The CIDR block of the SNAT rule conflicts with the network.	Either network_id or cidr must be specified.Both can not be specified at the same time	Do not specify the **Cidr** and **Network_id** fields at the same time when configuring an SNAT rule.
	400	NAT.0203	Invalid CIDR block.	cidr is invalid, make sure it's format is correct.	Enter a valid CIDR block, for example, 192.168.0.0/24.
	400	NAT.0204	Invalid rule type.	source_type and network_id is incompatible.	If an SNAT rule is configured for a VPC, the parameter **Source_Type** is optional or set its value to **0**. If an SNAT rule is configured for a Direct Connect connection, **Source_type** must be set to **1**.
	400	NAT.0205	The CIDR block must be a subset of the VPC subnet CIDR block.	cidr must be a subset of subnet's cidr.	If an SNAT rule is configured for a VPC, the CIDR block must be the VPC subnet CIDR block. For example, if the subnet is 192.168.0.0/24, the CIDR block can be 192.168.0.0/25.
	400	NAT.0206	The CIDR block conflicts with the subnet CIDR block.	cidr conflicts with subnet's cidr.	If an SNAT rule is configured for a Direct Connect connection, the CIDR block cannot conflict with the VPC subnet CIDR block.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0207	The CIDR block conflicts with the existing one.	cidr in the request conflicts with cidrs of existing rules.	Enter a CIDR block that does not conflict with existing ones.
	400	NAT.0208	The rule already exists.	Snat rule for network %(network)s exists.	Select a subnet that has not been configured with an SNAT rule.
	404	NAT.0209	The SNAT rule does not exist.	No Snat Rule exist with id %(id)s	Check whether the ID of the SNAT rule exists.
	400	NAT.0210	The floating IP address UUID of the SNAT rule is invalid.	Invalid input for floating_ip_id. Reason: \'%(fip)s\' is not a valid UUID.	Enter a valid UUID.
	400	NAT.0210	The public IP address ID of an SNAT rule cannot be a null string.	Invalid value for public ip id.	Enter a valid ID.
	400	NAT.0211	The number of EIPs associated with the SNAT rule exceeds the upper limit.	%(limit)s EIP has been associated to this SNAT rule's EIP pool, no more is allowed.	The number of EIPs associated with the SNAT rule exceeds the upper limit. For details, see the *NAT Gateway API Reference*.
	400	NAT.0212	Failed to bind or unbind the floating IP address to or from the SNAT rule.	SNAT Rule %(rule)s Associated or disassociate EIP %(fip)s Failed."	Failed to bind or unbind the floating IP address to or from the SNAT rule.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution

DnatRule	400	NAT.0301	Incorrect DNAT rule parameter.	get dnatRules error limit is invalid.
				//get dnatrules error marker is invalid.
				//endpoint is empty.
				//DnatRule id invalid.
				//VPC ID is invalid.
				//Request is null.
				//DnatRule id invalid.
				// internal_service_port_range' out of range(1-65535).
				// internal_service_port_range': invalid format.
				// internal_service_port_range': param is null.
				//'internal_service_port_range' and 'external_service_port_range' must be equal.
				//for non-all port rule,the protocol can not be any
				//param nat_gateway_id is null in request body
				//param floating_ip_id is
					Enter the correct parameter or contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
				null in request body //param xxxxxxx is null in request body	
	400	NAT.0302	Invalid DNAT rule protocol.	Dnat rule protocol %(protocol)s not supported.Only protocol values %(values)s and integer representations [6, 17, 0] are supported.	Configure a valid protocol. The value range can be **6**, **17**, or **0**, corresponding to protocols **TCP**, **UDP**, and **ANY**, respectively.
	400	NAT.0303	Invalid DNAT rule port.	Invalid value for port %(port)s	Configure a valid internal port and external port. The value ranges from 0 to 65535.
	400	NAT.0304	The internal network information of the DNAT rule conflicts with the existing one.	The port_id, private_ip, internal port and protocol specified have been occupied.	Enter the VM port ID, private IP address, or internal port that does not conflict with the existing one.
	400	NAT.0305	The external network information of the DNAT rule conflicts with the existing one.	The floating ip, external port and protocol specified have been occupied.	Enter the floating IP address ID, external port, or protocol that does not conflict with the existing one.
	400	NAT.0306	The request information of the DNAT rule is incorrect when **Port Type** is set to **All ports**.	The external port equals 0 and internal port equals 0 and protocol equals any must satisfied at the same time.	Set both the internal port and the external port to **0** and the protocol is **ANY**.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0307	The port ID of the DNAT rule conflicts with that of an existing DNAT rule.	The port_id already existing dnat allport rules or dnat_rules, can no longer create dnat rules or dnat allport rules.	Change the VM port ID to create or modify the DNAT rule.
	400	NAT.0308	The private IP address of the DNAT rule conflicts with that of an existing DNAT rule.	The private_ip already existing dnat allport rules or dnat rules, can no longer create dnat rules or dnat allport rules.	The private IP address conflicts with the existing DNAT rule. Change the private IP address or modify the DNAT rule.
	400	NAT.0318	The DNAT rule has been frozen and cannot be modified.	DNAT rule is frozen, can no longer update.	Check whether the floating IP address bound to the DNAT rule is in arrears or whether the user account is in arrears.
	400	NAT.0309	The maximum number of DNAT rules allowed to be bound has been reached.	%(limit)s DNAT rules has been associated to this NAT Gateway, no more is allowed	The maximum number of DNAT rules allowed to be associated with the NAT gateway has been reached.
	400	NAT.0317	The DNAT rule contains mutually exclusive parameters.	The port_id and private_ip exist at the same time and value is not empty, but at least one value is empty.	The VM port ID and private IP address cannot be configured at the same time.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0310	The parameters required by the DNAT rule are missing.	The port_id and private_ip values are both empty, at least one value is not empty.	Configure the VM port ID and private IP address.
	400	NAT.0311	Invalid private IP address of the DNAT rule.	The private ip address is not legal.	Configure a valid private IP address.
	400	NAT.0312	This virtual IP address is not supported.	The virtual IP address is not supported.	Configure a valid private IP address.
	400	NAT.0313	The maximum number of DNAT rules allowed to be bound has been reached.	%(limit)s DNAT rules has been associated to this Floating IP, no more is allowed	The maximum number of DNAT rules allowed to be associated with a floating IP address has been reached.
	400	NAT.0314	The maximum number of DNAT rules allowed to be created in batches exceeds the upper limit.	batch create dnat rules max limit: %(limit)s	The maximum number of DNAT rules allowed to be created in batches exceeds the upper limit.
	400	NAT.0315	Invalid VM port ID of the DNAT rule.	Port %(port)s is not a valid port.	Configure a valid VM port ID.
	400	NAT.0316	The value of **VtepIp** must be specified.	Vtep_ip is Null.	Delete and recreate the DNAT rule or contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	404	NAT.0319	The DNAT rule does not exist.	No Dnat Rule exist with id %(id)s	The DNAT rule does not exist.
EIP	400	NAT.0401	The EIP is frozen.	Floating Ip %(fip)s is frozen.	Select an EIP that has not been frozen.
	400	NAT.0402	The EIP has been associated with a port.	Floating Ip %(fip)s has associated with port %(port)s.	Select an EIP that has not been bound to any other object. For example, if an EIP has been bound to an ECS, it cannot be bound to a NAT gateway.
	400	NAT.0403	The EIP has been used by the SNAT rule.	There is a duplicate EIP %(fips)s in SNAT rule.	Select another EIP.
	400	NAT.0404	The EIP has been associated with a NAT gateway.	Floating Ip %(fip)s has used by nat gateway %(nat_gateway)s.	The EIP has been bound to a NAT gateway. Select another one.
	400	NAT.0405	The EIP is in use.	Floating Ip %(fip)s has been occupied.	The EIP has been bound to a NAT gateway. Select another one.
	400	NAT.0408	An EIP cannot be associated with a DNAT rule and another DNAT rule with **Port Type** set to **All ports** at the same time.	Floating Ip %(fip)s can not be associated with both DNAT rule and DNAT all port rule.	Do not associate an EIP with a DNAT rule and another DNAT rule with **Port Type** set to **All ports** at the same time.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	NAT.0409	An EIP cannot be associated with an SNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.	Floating Ip % (fip)s can not be associated with both SNAT rule and DNAT all port rule.	Do not associate an EIP with an SNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.
	400	NAT.0407	The EIP has been associated with a rule.	Floating Ip % (fip)s is used by other rules	Select an EIP that is not in use.

7.3 Error Code Description v2.0

Background Information

- An error code returned by an API does not correspond to one error message. The following table lists only common error messages.
- Most NAT Gateway APIs are asynchronous. Some error codes are displayed in the returned messages for task viewing requests. HTTP status codes may not be accurate.
- The NAT Gateway service is strongly dependent on other services, such as network and storage. When error messages are provided for the NAT Gateway-depended services, contact technical support for troubleshooting.

Error Codes

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
Public	400	VPC.0002	The AZ is left blank.	Availability zone Name is null.	Verify whether the **availability_zone** field in the request body for creating a subnet is left blank.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	404	VPC.0003	The VPC does not exist.	VPC does not exit.	Check whether the VPC ID is correct or whether the VPC exists under the tenant.
	400	VPC.0004	The status of the VPC is abnormal.	VPC is not active, please try later.	Try again later or contact technical support.
	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.
	401	VPC.0009	Real-name authentication failed.	real-name authentication failed.	Contact technical support.
	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.
	403	VPC.0010	Insufficient permissions to make calls to the underlying system.	Rules on xx by ** disallowed by policy	Obtain the required permissions.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	403	VPC.2201	Insufficient fine-grained permissions.	Policy does not allow <x:x:x> to be performed	Obtain the required permissions.
	400	VPC.0014	The enterprise project is unavailable.	This enterpriseProject status is disabled.	Use the ID of another available enterprise project.
	400	VPC.0011	Invalid enterprise project ID.	EnterpriseProjectId is invalid	Enter a valid enterprise project ID.
	400	VPC.2048	Invalid timestamp.	Invalid value for created_at % (timestamp)s.	Enter the time in the correct format.
	400	VPC.2002	Invalid request parameters.	Invalid parameters.	Enter the correct parameter.
	400	VPC.2010	The default route already exists.	The router % (router_id)s has default route.	The router has a default route. Delete the default route and then create a NAT gateway.
	400	VPC.2011	The router does not exist.	The router % (router_id)s does not exist.	Check whether the entered router ID is correct.
	400	VPC.2009	The network does not exist.	Network % (network_id)s does not exist.	Enter a valid network ID.
	400	VPC.2016	The rule has not been deleted.	Rule has not been deleted.	Contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2049	The database is abnormal.	DB Error	Contact technical support.
	400	VPC.2013	The subnet is not connected to the virtual router.	Router %(router)s has no port for subnet %(subnet)s.	Add the subnet to the router port.
	400	VPC.2019	The resource is in use.	Resource %(res_type)s %(res)s is used by %(user_type)s %(user)s	Contact technical support.
	400	VPC.2008	The subnet does not exist in the VPC.	Network %(network)s does not contain any IPv4 subnet	Contact technical support.
	400	VPC.2012	The VPC already has a NAT gateway.	The router %(router_id)s already has nat gateway.	Select another VPC.

NAT Gateway	400	VPC.2000	NAT gateway request error.	Lack of user authority.
				//request is null.
				//endpoint is empty.
				// resource type is invalid. //create natgateway request is null. //update natgateway request is null
					Contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2030	The system is busy. Please try again later.	The system is busy. Please try again later.	Try again later.

	400	VPC.2001	Incorrect NAT gateway parameter.	Request is invalid. // NatGateway id invalid.
				// the enterprise project id is unsupported.
				// the enterprise project id in request is invalid. // parameter is null.
				// tags is invalid.
				// get natgateways error limit is invalid.
				//get natgateways error marker is invalid.
				//Only admin user can do this action. //Parameters are invalid, check them and try.
					Enter the correct parameter or contact technical support.

	400	VPC.2004	The NAT gateway is not activated.	NatGateway %(nat_gateway_id)s is not ACTIVE.	Check the gateway status. If the gateway is not in the running state for a long time, contact technical support.
	400	VPC.2005	The NAT gateway is not in the UP state.	NatGateway %(nat_gateway_id)s is not UP.	The gateway may be frozen due to arrears.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2006	The NAT gateway is frozen.	NatGateway %(nat_gateway_id)s is frozen.cannot update	The gateway may be frozen due to arrears and cannot be updated.
	400	VPC.2007	The NAT gateway does not exist.	NatGateway %(nat_gateway_id)s does not exist.	The NAT gateway does not exist.
	400	VPC.2050	Concurrent operation conflicts.	Concurrent conflict requests found	Contact technical support.
	400	VPC.2051	Failed to create the internal port of the NAT gateway.	Create NG Port failed.	Internal error. Contact technical support.
	400	VPC.2052	Failed to bind the internal port to the NAT gateway.	NG Port %(port)s is unbound.	Internal error. Contact technical support.
	400	VPC.2053	The NAT gateway does not support IPv6.	NatGateway does not support IPv6.	The NAT gateway cannot be bound to an IPv6 EIP.
	400	VPC.2045	An error occurred when selecting the gateway node.	Get Nat gateway host failed	Contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2046	Failed to obtain the IP address of the gateway node.	Get Nat gateway agent local_ip failed	Contact technical support.
	400	VPC.2047	Failed to obtain the VPC route table.	Get RouteTable %(router_id)s failed.	Contact technical support.
	400	VPC.2012	The router already has a NAT gateway.	The router %(router_id)s already has nat gateway.	Select a router that has not been bound to a NAT gateway.

SNAT Rule	400	VPC.2014	Incorrect SNAT rule parameter.	Endpoint is null or empty. //Endpoint is Invalid.
				//Request is null. // natGatewayId is invalid. //SnatRule id invalid. // NatGatewayId is invalid.
				//Invalid value for public ip id.
				//Endpoint is null.
				//request is null.
				//Query SnatRules list error marker is invalid.
					Enter the correct parameter or contact technical support.

	400	VPC.2031	The CIDR of the SNAT rule conflicts with the network.	Either network_id or cidr must be specified.Both cannot be specified at the same time	Do not specify the **Cidr** and **Network_id** fields at the same time when configuring an SNAT rule.
	400	VPC.2032	Invalid CIDR block.	cidr is invalid, make sure its format is correct.	Enter a valid CIDR block, for example, 192.168.0.0/24.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2033	Invalid rule type.	source_type and network_id is incompatible.	If an SNAT rule is configured for a VPC, the parameter **Source_Type** is optional or set its value to **0**. If an SNAT rule is configured for a Direct Connect connection, **Source_type** must be set to **1**.
	400	VPC.2034	The CIDR block must be a subset of the VPC subnet CIDR block.	cidr must be a subset of subnet's cidr.	If an SNAT rule is configured for a VPC, the CIDR block must be the VPC subnet CIDR block. For example, if the subnet is 192.168.0.0/24, the CIDR block can be 192.168.0.0/25.
	400	VPC.2035	The CIDR block conflicts with the subnet CIDR block.	cidr conflicts with subnet's cidr.	If an SNAT rule is configured for a Direct Connect connection, the CIDR block cannot conflict with the VPC subnet CIDR block.
	400	VPC.2036	The CIDR block conflicts with the existing one.	cidr in the request conflicts with cidrs of existing rules.	Enter a CIDR block that does not conflict with existing ones.
	400	VPC.2018	The rule already exists.	Snat rule for network % (network)s exists.	Select a subnet that is not configured with an SNAT rule.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2042	The EIP has been used by the SNAT rule.	There is a duplicate EIP %(fips)s in SNAT rule.	Select another EIP.
	400	VPC.2044	The public IP address UUID of the SNAT rule is invalid.	Invalid input for floating_ip_id. Reason: \'%(fip)s\' is not a valid UUID.	Enter a valid UUID.
	400	VPC.2040	The public IP address ID of an SNAT rule cannot be left blank.	Invalid value for public ip id.	Enter a valid ID.
	400	VPC.2039	The number of EIPs associated with the SNAT rule exceeds the upper limit.	%(limit)s EIP has been associated to this SNAT rules' EIP pool, no more is allowed.	The number of EIPs associated with the SNAT rule exceeds the upper limit. For details, see the *NAT Gateway API Reference*.

DNAT Rule	400	VPC.2020	Incorrect DNAT rule parameter.	get dnatRules error limit is invalid.
				//get dnatrules error marker is invalid.
				//endpoint is empty.
				//DnatRule id invalid.
				//VPC ID is invalid.
				//Request is null.
				//DnatRule id invalid.
				//DnatRule natGatewayId id invalid.
					Enter the correct parameter or contact technical support.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2054	Invalid DNAT rule protocol.	Dnat rule protocol %(protocol)s not supported.Only protocol values %(values)s and integer representations [6, 17, 0] are supported.	Configure a valid protocol. The value range can be **6**, **17**, or **0**, corresponding to protocols **TCP**, **UDP**, and **ANY**, respectively.
	400	VPC.2069	Invalid DNAT rule port.	Invalid value for port %(port)s	Configure a valid internal port and external port. The value ranges from 0 to 65535.
	400	VPC.2023	The internal network information of the DNAT rule conflicts with the existing one.	The port_id, private_ip, internal port, and protocol specified have been occupied.	Enter the VM port ID, private IP address, or internal port that does not conflict with the existing one.
	400	VPC.2024	The external network information of the DNAT rule conflicts with the existing one.	The floating ip, external port and protocol specified have been occupied.	Enter the floating IP address ID, external port, or protocol that does not conflict with the existing one.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2070	The request information of the DNAT rule is incorrect when **Port Type** is set to **All ports**.	The external port equals 0 and internal port equals 0 and protocol equals any must satisfied at the same time.	Set both the internal port and the external port to **0** and the protocol is **ANY**.
	400	VPC.2027	The port ID of the DNAT rule conflicts with that of an existing DNAT rule.	The port_id already existing dnat allport rules or dnat_rules, can no longer create dnat rules or dnat allport rules.	Change the VM port ID to create or modify the DNAT rule.
	400	VPC.2028	The private IP address of the DNAT rule conflicts with that of an existing DNAT rule.	The private_ip already existing dnat allport rules or dnat rules, can no longer create dnat rules or dnat allport rules.	The private IP address conflicts with the existing DNAT rule. Change the private IP address or modify the DNAT rule.
	400	VPC.2029	The DNAT rule has been frozen and cannot be modified.	DNAT rule is frozen, can no longer update.	Check whether the floating IP address bound to the DNAT rule is in arrears or whether the user account is in arrears.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2038	The maximum number of DNAT rules allowed to be bound has been reached.	%(limit)s DNAT rules has been associated to this NAT Gateway, no more is allowed	The maximum number of DNAT rules allowed to be associated with the NAT gateway has been reached.
	400	VPC.2055	The DNAT rule contains mutually exclusive parameters.	The port_id and private_ip exist at the same time and value is not empty, but at least one value is empty.	The VM port ID and private IP address cannot be configured at the same time.
	400	VPC.2056	The DNAT rule does not have required parameters configured.	The port_id and private_ip values are both empty, at least one value is not empty.	Configure the VM port ID and private IP address.
	400	VPC.2071	Invalid private IP address of the DNAT rule.	The private ip address is not legal.	Configure a valid private IP address.
	400	VPC.2037	This virtual IP address is not supported.	The virtual IP address is not supported.	Configure a valid private IP address.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2026	The maximum number of DNAT rules allowed to be bound has been reached.	%(limit)s DNAT rules has been associated to this Floating IP, no more is allowed	The maximum number of DNAT rules allowed to be associated with a floating IP address has been reached.
	400	VPC.2057	The maximum number of DNAT rules allowed to be created in batches exceeds the upper limit.	batch create dnat rules max limit: %(limit)s	The maximum number of DNAT rules allowed to be created in batches exceeds the upper limit.
	400	VPC.2022	Invalid VM port ID of the DNAT rule.	Port %(port)s is not a valid port.	Configure a valid VM port ID.
	400	VPC.2058	The value of **VtepIp** must be specified.	Vtep_ip is Null.	Contact technical support.
	400	VPC.2075	The description contains more than 255 characters.	Enter a maximum of 255 characters.	Enter a maximum of 255 characters.
EIP	400	VPC.2059	The EIP is frozen.	Floating Ip %(fip)s is frozen.	Select an EIP that has not been frozen.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2060	The EIP has been associated with a port.	Floating Ip %(fip)s has associated with port %(port)s.	Select an EIP that has not been bound to any other object. For example, if an EIP has been bound to an ECS, it cannot be bound to a NAT gateway.
	400	VPC.2061	The EIP has been associated with a NAT gateway.	Floating Ip %(fip)s has used by nat gateway %(nat_gateway)s.	The EIP has been bound to a NAT gateway. Select another one.
	400	VPC.2062	The EIP is in use.	Floating Ip %(fip)s has been occupied.	The EIP has been bound to a NAT gateway. Select another one.
	400	VPC.2074	An EIP cannot be associated with an SNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.	Floating Ip %(fip)s cannot be associated with both SNAT rule and DNAT all port rule.	Do not associate an EIP with an SNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.

Module	HTTP Status Code	Error Code	Description	Error Message	Solution
	400	VPC.2073	An EIP cannot be associated with a DNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.	Floating Ip %(fip)s cannot be associated with both DNAT rule and DNAT all port rule.	Do not associate an EIP with a DNAT rule and a DNAT rule with **Port Type** set to **All ports** at the same time.
	400	VPC.2043	The EIP has been associated with a rule.	Floating Ip %(fip)s is used by other rules	Select an EIP that is not in use.

7.4 Obtaining a Project ID

Scenarios

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

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

Obtain the Project ID by Calling an API

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

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

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

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
```



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


Obtain a Project ID from the Console

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

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

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

7.5 Resource Status Description

Table 7-2 Resource status description

Status	Description
ACTIVE	The resource status is normal.
PENDING_CREATE	The resource is being created.
PENDING_UPDATE	The resource is being updated.
PENDING_DELETE	The resource is being deleted.
EIP_FREEZED	The EIP of the resource is frozen.
INACTIVE	The resource status is abnormal.

8 Out-of-Date APIs

8.1 API v2.0

8.1.1 NAT Gateway Service

8.1.1.1 Creating a NAT Gateway

Function

This API is used to create a NAT gateway.

URI

POST /v2.0/nat_gateways

Request

Table 8-1 describes the request parameters.

Table 8-1 Request parameter

Parameter	Mandatory	Type	Description
nat_gateway	Yes	Object	Specifies the NAT gateway object. For details, see Table 8-2.

Table 8-2 Description of the **nat_gateway** field

Parameter	Mandatory	Type	Description
tenant_id	No	String	Specifies the project ID.
name	Yes	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	Yes	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	Yes	String	Specifies the VPC ID.
internal_network_id	Yes	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.

Response

Table 8-3 lists response parameters.

Table 8-3 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see Table 8-4.

Table 8-4 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the NAT gateway status. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the NAT gateway is frozen. • The value can be: – **true**: The NAT gateway is unfrozen. – **false**: The NAT gateway is frozen.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request

```
POST https://{Endpoint}/v2.0/nat_gateways
{
   "nat_gateway": {
      "name": "nat_001",
      "description": "my nat gateway 01",
      "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
      "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
      "spec": "1"
   }
}
```

- Example response

```
{
   "nat_gateway": {
       "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
       "status": "PENDING_CREATE",
       "description": "my nat gateway 01",
       "admin_state_up": true,
       "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
       "created_at": "2017-11-18 07:34:32.203044",
       "spec": "1",
       "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
       "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
       "name": "nat_001"
   }
}
```

Status Codes

See **Status Codes**.

8.1.1.2 Querying NAT Gateways

Function

This API is used to query a NAT gateway list. Unless otherwise specified, exact match is applied.

URI

GET /v2.0/nat_gateways

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 8-5 Parameter description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the NAT gateway ID.

Parameter	Mandatory	Type	Description
limit	No	Integer	Specifies the number of records on each page.
tenant_id	No	String	Specifies the project ID.
name	No	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	No	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	No	String	Specifies the VPC ID.
internal_network_id	No	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	No	String	• Specifies the NAT gateway status. • For details about all its values, see **Table 7-2**.
admin_state_up	No	Boolean	• Specifies whether the NAT gateway is frozen. • The value can be: – **true**: The NAT gateway is unfrozen. – **false**: The NAT gateway is frozen.
created_at	No	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Request

None

Response

Table 8-6 lists response parameters.

Table 8-6 Response parameter

Parameter	Type	Description
nat_gateways	List (NAT gateways)	Specifies the NAT gateway objects. For details, see Table 8-7.

Table 8-7 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.

Parameter	Type	Description
status	String	• Specifies the NAT gateway status. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the NAT gateway is frozen. • The value can be: – **true**: The NAT gateway is unfrozen. – **false**: The NAT gateway is frozen.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/nat_gateways?limit=10
  ```
- Example response
  ```
  {
      "nat_gateways": [
          {
              "router_id": "b1d81744-5165-48b8-916e-e56626feb88f",
              "status": "ACTIVE",
              "description": "",
              "admin_state_up": true,
              "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
              "created_at": "2017-11-15 14:50:39.505112",
              "spec": "2",
              "internal_network_id": "5930796a-6026-4d8b-8790-6c6bfc9f87e8",
              "id": "a253be25-ae7c-4013-978b-3c0785eccd63",
              "name": "wj3"
          },
          {
              "router_id": "305dc52f-13dd-429b-a2d4-444a1039ba0b",
              "status": "ACTIVE",
              "description": "",
              "admin_state_up": true,
              "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
              "created_at": "2017-11-17 07:41:07.538062",
              "spec": "2",
              "internal_network_id": "fc09463b-4ef8-4c7a-93c8-92d9ca6daf9d",
              "id": "e824f1b4-4290-4ebc-8322-cfff370dbd1e",
              "name": "lyl001"
          }
      ]
  }
  ```

Status Codes

See **Status Codes**.

8.1.1.3 Querying Details About a Specified NAT Gateway

Function

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

URI

GET /v2.0/nat_gateways/{nat_gateway_id}

Table 8-8 Parameter description

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.

Request

None

Response

Table 8-9 lists response parameters.

Table 8-9 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see Table 8-10.

Table 8-10 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.

Parameter	Type	Description
spec	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the NAT gateway status. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the NAT gateway is frozen. • The value can be: – **true**: The NAT gateway is unfrozen. – **false**: The NAT gateway is frozen.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
  ```
- Example response
  ```
  {
     "nat_gateway": {
         "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
         "status": "ACTIVE",
         "description": "my nat gateway 01",
         "admin_state_up": true,
         "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
         "created_at": "2017-11-18 07:34:32.203044",
         "spec": "1",
         "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
         "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
         "name": "nat_001"
     }
  }
  ```


Status Codes

See **Status Codes**.

8.1.1.4 Updating a NAT Gateway

Function

This API is used to update a NAT gateway.

NOTE

admin_state_up = True & status = "ACTIVE" can be updated. The name, description, and type of a NAT gateway can be updated.

URI

PUT /v2.0/nat_gateways/{nat_gateway_id}

Table 8-11 Parameter description

Parameter	Type	Mandatory	Description
nat_gateway_id	String	Yes	Specifies the NAT gateway ID.

Request

Table 8-12 describes the request parameters.

Table 8-12 Request parameter

Parameter	Mandatory	Type	Description
nat_gateway	Yes	Object	Specifies the NAT gateway object. For details, see **Table 8-13**. Mandatory field: None. Only the **name**, **description**, and **spec** fields can be updated. At least one attribute must be specified for the NAT gateway to be updated.

Table 8-13 Description of the **nat_gateway** field

Parameter	Mandatory	Type	Description
name	No	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	No	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	No	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.

Response

Table 8-14 lists response parameters.

Table 8-14 Response parameter

Parameter	Type	Description
nat_gateway	Object	Specifies the NAT gateway object. For details, see Table 8-15.

Table 8-15 Description of the **nat_gateway** field

Parameter	Type	Description
id	String	Specifies the NAT gateway ID.
tenant_id	String	Specifies the project ID.

Parameter	Type	Description
name	String	Specifies the NAT gateway name. You can enter up to 64 characters. The name can contain only digits, letters, underscores (_), and hyphens (-).
description	String	Provides supplementary information about the NAT gateway. You can enter up to 255 characters.
spec	String	Specifies the NAT gateway type. The type can be: • **1**: small type, which supports up to 10,000 SNAT connections. • **2**: medium type, which supports up to 50,000 SNAT connections. • **3**: large type, which supports up to 200,000 SNAT connections. • **4**: extra-large type, which supports up to 1,000,000 SNAT connections.
router_id	String	Specifies the VPC ID.
internal_network_id	String	Specifies the network ID of the downstream interface (the next hop of the DVR) of the NAT gateway.
status	String	• Specifies the NAT gateway status. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the NAT gateway is frozen. • The value can be: – **true**: The NAT gateway is unfrozen. – **false**: The NAT gateway is frozen.
created_at	String	Specifies when the NAT gateway is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request

```
PUT https://{Endpoint}/v2.0/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
 {
   "nat_gateway": {
      "name": "new_name",
      "description": "new description",
      "spec": "1"
```



```
    }
}
```

- Example response

```
{
    "nat_gateway": {
        "router_id": "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
        "status": "ACTIVE",
        "description": "new description",
        "admin_state_up": true,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:34:32.203044",
        "spec": "1",
        "internal_network_id": "89d66639-aacb-4929-969d-07080b0f9fd9",
        "id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "name": "new_name"
    }
}
```

Status Codes

See **Status Codes**.

8.1.1.5 Deleting a NAT Gateway

Function

This API is used to delete a NAT gateway.

URI

DELETE /v2.0/nat_gateways/{nat_gateway_id}

Table 8-16 Parameter description

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.

Request

None

Response

None

Examples

- Example request

```
DELETE https://{Endpoint}/v2.0/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
```

- Example response

```
None (STATUS CODE 204)
```


Status Code

See **Status Codes**.

8.1.2 SNAT Rules

8.1.2.1 Creating an SNAT Rule

Function

This API is used to create an SNAT rule.

NOTE

You can create an SNAT rule only when **status** of the NAT gateway is set to **ACTIVE** and **admin_state_up** of the NAT gateway administrator to **True**.

URI

POST /v2.0/snat_rules

Request

Table 8-17 describes the request parameters.

Table 8-17 Request parameter

Parameter	Mandatory	Type	Description
snat_rule	Yes	Object	Specifies the SNAT rule object. For details, see Table 8-18.

Table 8-18 Description of the **snat_rule** field

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
network_id	No	String	Specifies the network ID used by the SNAT rule. Either this parameter or **cidr** must be specified.

Parameter	Mandatory	Type	Description
cidr	No	String	Specifies CIDR, which can be in the format of a network segment or a host IP address. This parameter and **network_id** are alternative. If **Source_type** is set to **0**, the CIDR block must be a subset of the VPC subnet CIDR block. If **Source_type** is set to **1**, **cidr** must be a CIDR block of Direct Connect connection.
source_type	No	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, default value **0** (VPC) is used.
floating_ip_id	Yes	String	Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). The maximum length of the ID is 4096 bytes. The number of EIP IDs cannot exceed 20.

Response

Table 8-19 lists response parameters.

Table 8-19 Response parameter

Parameter	Type	Description
snat_rule	Object	Specifies the SNAT rule object. For details, see Table 8-20.

Table 8-20 Description of the **snat_rule** field

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.

Parameter	Type	Description
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the SNAT rule is frozen. • The value can be: – **true**: The SNAT rule is unfrozen. – **false**: The SNAT rule is frozen.
created_at	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  a. Configure parameter **network_id** in a VPC.

```
POST https://{Endpoint}/v2.0/snat_rules
{
   "snat_rule": {
      "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
      "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
      "source_type":0,
      "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a"
   }
}
```


b. Configure parameter **cidr** in a VPC.

```
POST https://{Endpoint}/v2.0/snat_rules
{
  "snat_rule": {
       "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
       "cidr": "192.168.1.10/32",
       "source_type":0,
       "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a"
    }
 }
```

c. Configure parameter **cidr** over a Direct Connect connection.

```
POST https://{Endpoint}/v2.0/snat_rules
 {
    "snat_rule": {
       "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
       "cidr": "172.30.0.0/24",
       "source_type":1,
       "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a"
    }
 }
```

- Example response

a. Response to the request for specifying the **network_id** for a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "cidr": null,
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
    }
}
```

b. Response to the request for specifying the CIDR block in a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "cidr": "192.168.1.10/32",
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
    }
}
```

c. Response to the request for specifying the CIDR block in a VPC

```
{
    "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "PENDING_CREATE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "cidr": "172.30.0.0/24",

        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
```



```
    }
}
```

Status Codes

See **Status Codes**.

8.1.2.2 Querying SNAT Rules

Function

This API is used to query an SNAT rule list.

URI

GET /v2.0/snat_rules

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 8-21 Parameter description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the SNAT rule ID.
limit	No	Integer	Specifies the number of records on each page.
tenant_id	No	String	Specifies the project ID.
nat_gateway_id	No	String	Specifies the NAT gateway ID.
network_id	No	String	Specifies the network ID used by the SNAT rule.
cidr	No	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	No	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, default value **0** (VPC) is used.
floating_ip_id	No	String	• Specifies the EIP ID. • Maximum length: 4,096 characters

Parameter	Mandatory	Type	Description
floating_ip_address	No	String	• Specifies the EIP. • Maximum length: 1,024 characters
status	No	String	• Specifies the status of the SNAT rule. • For details about all its values, see Table 7-2.
admin_state_up	No	Boolean	• Specifies whether the SNAT rule is frozen. • The value can be: – **true**: The SNAT rule is unfrozen. – **false**: The SNAT rule is frozen.
created_at	No	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Request

None

Response

Table 8-22 lists response parameters.

Table 8-22 Response parameter

Parameter	Type	Description
snat_rules	List (SNAT rules)	Specifies the SNAT rule objects. For details, see Table 8-23.

Table 8-23 Description of the **snat_rule** field

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.

Parameter	Type	Description
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes. • The number of EIP IDs cannot exceed 20.
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the SNAT rule is frozen. • The value can be: – **true**: The SNAT rule is unfrozen. – **false**: The SNAT rule is frozen.
created_at	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/snat_rules?limit=10
  ```
- Example response
  ```
  {
    "snat_rules": [
  ```



```
    {
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "network_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "cidr": "null",
        "source_type":0,
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.242"
      },
      {
        "floating_ip_id": "6e496fba-abe9-4f5e-9406-2ad8c809ac8c",
        "status": "ACTIVE",
        "nat_gateway_id": "e824f1b4-4290-4ebc-8322-cfff370dbd1e",
        "admin_state_up": true,
        "network_id": "97e89905-f9c8-4ae3-9856-392b0b2fbe7f",
        "cidr": "null",
        "source_type":0,
        "tenant_id": "abc",
        "created_at": "2017-11-17 07:43:44.830845",
        "id": "4a1a10d7-0d9f-4846-8cda-24cffeffef5c",
        "floating_ip_address": "5.21.11.142"
      }
    ]
}
```


Status Codes

See **Status Codes**.

8.1.2.3 Querying Details About a Specified SNAT Rule

Function

This API is used to query details about a specified SNAT rule.

URI

GET /v2.0/snat_rules/{snat_rule_id}

Table 8-24 Parameter description

Parameter	Mandatory	Type	Description
snat_rule_id	Yes	String	Specifies the SNAT rule ID.

Request

None

Response

Table 8-25 lists response parameters.

Table 8-25 Response parameter

Parameter	Type	Description
snat_rule	Object	Specifies the SNAT rule object. For details, see **Table 8-26**.

Table 8-26 Description of the **snat_rule** field

Parameter	Type	Description
id	String	Specifies the SNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
network_id	String	Specifies the network ID used by the SNAT rule.
cidr	String	Specifies a subset of the VPC subnet CIDR block or a CIDR block of Direct Connect connection.
source_type	Integer	**0**: Either **network_id** or **cidr** can be specified in a VPC. **1**: Only **cidr** can be specified over a Direct Connect connection. If no value is entered, default value **0** (VPC) is used.
floating_ip_id	String	• Specifies the EIP ID. Multiple EIP IDs must be separated using commas (,). • The maximum length of the ID is 4096 bytes. • The number of EIP IDs cannot exceed 20.

Parameter	Type	Description
floating_ip_address	String	• Specifies the EIP. Multiple EIPs must be separated using commas (,). • The maximum length is 1024 bytes.
status	String	• Specifies the status of the SNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the SNAT rule is frozen. • The value can be: – **true**: The SNAT rule is unfrozen. – **false**: The SNAT rule is frozen.
created_at	String	Specifies when the SNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
  ```
- Example response
  ```
  {
     "snat_rule": {
        "floating_ip_id": "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
        "status": "ACTIVE",
        "nat_gateway_id": "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        "admin_state_up": true,
        "network_id": "eaad9cd6-2372-4be1-9535-9bd37210ae7b",
        "cidr": "null",
        "source_type":0,
        "tenant_id": "27e25061336f4af590faeabeb7fcd9a3",
        "created_at": "2017-11-18 07:54:21.665430",
        "id": "5b95c675-69c2-4656-ba06-58ff72e1d338",
        "floating_ip_address": "5.21.11.226"
     }
  }
  ```

Status Codes

See **Status Codes**.

8.1.2.4 Deleting an SNAT Rule

Function

This API is used to delete an SNAT rule.

URI

DELETE /v2.0/snat_rules/{snat_rule_id}

Table 8-27 Parameter description

Parameter	Mandatory	Type	Description
snat_rule_id	Yes	String	Specifies the SNAT rule ID.

Request

None

Response

None

Examples

- Example request
  DELETE https://{Endpoint}/v2.0/snat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
- Example response
  None (STATUS CODE 204)

Status Code

See **Status Codes**.

8.1.3 DNAT Rules

8.1.3.1 Creating a DNAT Rule

Function

This API is used to create a DNAT rule.

NOTE

You can create a DNAT rule only when **status** of the NAT gateway is **ACTIVE** and **admin_state_up** of the NAT gateway administrator is **True**. Specify either **port_id** or **private_ip** at a time. If you are going to create a DNAT rule that allows traffic to and from all ports of a server and an EIP, set **internal_service_port** to **0**, **external_service_port** to **0**, and **protocol** to **any**.

URI

POST /v2.0/dnat_rules

Request

Table 8-28 lists the request parameter.

Table 8-28 Request parameter

Parameter	Mandatory	Type	Description
dnat_rule	Yes	Object	Specifies the DNAT rule object. For details, see Table 8-29.

Table 8-29 Description of the **dnat_rule** field

Parameter	Mandatory	Type	Description
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
port_id	No	String	Specifies the port ID of an ECS or a BMS. This parameter and **private_ip** are alternative.
private_ip	No	String	Specifies the private IP address of a user, for example, the IP address of a VPC connected by a Direct Connect connection. You can specify either this parameter or **port_id**.
internal_service_port	Yes	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems. The value ranges from **0** to **65535**.
floating_ip_id	Yes	String	Specifies the EIP ID.
external_service_port	Yes	Integer	Specifies the port used by the floating IP address to provide services for external systems. The value ranges from **0** to **65535**.
protocol	Yes	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).

Response

Table 8-30 lists response parameter.

Table 8-30 Response parameter

Parameter	Type	Description
dnat_rule	Object	Specifies the DNAT rule object. For details, see Table 8-31.

Table 8-31 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS. This parameter is used in the VPC scenario. This parameter and **private_ip** are alternative.
private_ip	String	Specifies the private IP address, for example, the IP address of a Direct Connect connection. This parameter is used in the Direct Connect scenario. This parameter and **port_id** are alternative.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP address.
external_service_port	Integer	Specifies the port used by the floating IP address to provide services for external systems.
protocol	String	Specifies the protocol. Its value can be **tcp** (6), **udp** (17), or **any** (0).

Parameter	Type	Description
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the DNAT rule is frozen. • The value can be: – **true**: The DNAT rule is unfrozen. – **false**: The DNAT rule is frozen.
created_at	String	Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example requests
  a. Creating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
POST https://{Endpoint}/v2.0/dnat_rules
{
   "dnat_rule": {
      "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
      "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
      "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
      "internal_service_port": 993,
      "protocol": "tcp",
      "external_service_port": 242
   }
}
```

  b. Creating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**

```
POST https://{Endpoint}/v2.0/dnat_rules
{
   "dnat_rule": {
      "floating_ip_id": "Cf99c679-9f41-4dac-8513-9c9228e713e1",
      "nat_gateway_id": "Dda3a125-2406-456c-a11f-598e10578541",
      "private_ip": "192.168.1.100",
      "internal_service_port": 0,
      "protocol": "any",
      "external_service_port": 0
   }
}
```

- Example responses
  a. Response to the request for creating a DNAT rule with specified **internal_service_port** and **external_service_port**

```
{
   "dnat_rule": {
      "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
      "status": "ACTIVE",
```



```
        "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
        "internal_service_port": 993,
        "protocol": "tcp",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242,
        "private_ip": ""
    }
}
```


b. Response to the request for creating a DNAT rule with both **internal_service_port** and **external_service_port** set to **0**


```
{
    "dnat_rule": {
        "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
        "status": "ACTIVE",
        "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
        "private_ip": "192.168.1.100",
        "internal_service_port": 0,
        "protocol": "any",
        "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
        "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.227",
        "external_service_port": 0
    }
}
```


Status Codes

See **Status Codes**.

8.1.3.2 Querying DNAT Rules

Function

This API is used to query a DNAT rule list.

URI

GET /v2.0/dnat_rules

NOTE

You can type the question mark (?) and ampersand (&) at the end of the URI to define multiple search criteria. All optional parameters can be filtered. For details, see the example request.

Table 8-32 Parameter description

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
limit	Integer	Specifies the number of records on each page.

Parameter	Type	Description
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address, for example, the IP address of a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP.
external_service_port	Integer	Specifies the port used by the floating IP address to provide services for external systems.
protocol	String	Specifies the protocol type. Currently, TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Resource Status Description**.
admin_state_up	Boolean	• Specifies whether the DNAT rule is frozen. • The value can be: – **true**: The DNAT rule is unfrozen. – **false**: The DNAT rule is frozen.
created_at	String	Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Request

None

Response

Table 8-33 lists response parameters.

Table 8-33 Response parameter

Parameter	Type	Description
dnat_rules	Array(Object)	Specifies the DNAT rule objects. For details, see **Table 8-34**.

Table 8-34 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address, for example, the IP address of a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP.
external_service_port	Integer	Specifies the port used by the floating IP address to provide services for external systems.
protocol	String	Specifies the protocol type. Currently, TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.

Parameter	Type	Description
admin_state_up	Boolean	• Specifies whether the DNAT rule is frozen. • The value can be: – **true**: The DNAT rule is unfrozen. – **false**: The DNAT rule is frozen.
created_at	String	Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/dnat_rules
  ```
- Example response
  ```
  {
     "dnat_rules": [
        {
           "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
           "status": "ACTIVE",
           "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
           "admin_state_up": true,
           "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
           "internal_service_port": 993,
           "protocol": "tcp",
           "tenant_id": "abc",
           "created_at": "2017-11-15 15:44:42.595173",
           "id": "79195d50-0271-41f1-bded-4c089b2502ff",
           "floating_ip_address": "5.21.11.226",
           "external_service_port": 242,
           "private_ip": ""
        },
        {
           "floating_ip_id": "cf99c679-9f41-4dac-8513-9c9228e713e1",
           "status": "ACTIVE",
           "nat_gateway_id": "dda3a125-2406-456c-a11f-598e10578541",
           "admin_state_up": true,
           "port_id": "",
           "private_ip": "192.168.1.100",
           "internal_service_port": 0,
           "protocol": "any",
           "tenant_id": "abc",
           "created_at": "2017-11-16 15:44:42.595173",
           "id": "89195d50-0271-41f1-bded-4c089b2502ff",
           "floating_ip_address": "5.21.11.227",
                "external_service_port": 0
        }
     ]
  }
  ```


Status Codes

See **Status Codes**.

8.1.3.3 Querying Details About a Specified DNAT Rule

Function

This API is used to query details about a specified DNAT rule.

URI

GET /v2.0/dnat_rules/{dnat_rule_id}

Table 8-35 Parameter description

Parameter	Type	Mandatory	Description
dnat_rule_id	String	Yes	Specifies the DNAT rule ID.

Request

None

Response

Table 8-36 lists response parameters.

Table 8-36 Response parameter

Parameter	Type	Description
dnat_rule	Object	Specifies the DNAT rule object. For details, see **Table 8-37**.

Table 8-37 Description of the **dnat_rule** field

Parameter	Type	Description
id	String	Specifies the DNAT rule ID.
tenant_id	String	Specifies the project ID.
nat_gateway_id	String	Specifies the NAT gateway ID.

Parameter	Type	Description
port_id	String	Specifies the port ID of an ECS or a BMS.
private_ip	String	Specifies the private IP address, for example, the IP address of a Direct Connect connection.
internal_service_port	Integer	Specifies the port used by ECSs or BMSs to provide services for external systems.
floating_ip_id	String	Specifies the EIP ID.
floating_ip_address	String	Specifies the EIP.
external_service_port	Integer	Specifies the port used by the floating IP address to provide services for external systems.
protocol	String	Specifies the protocol type. Currently, TCP, UDP, and ANY are supported. The protocol number of TCP, UDP, and ANY are 6, 17, and 0, respectively.
status	String	• Specifies the status of the DNAT rule. • For details about all its values, see **Table 7-2**.
admin_state_up	Boolean	• Specifies whether the DNAT rule is frozen. • The value can be: – **true**: The DNAT rule is unfrozen. – **false**: The DNAT rule is frozen.
created_at	String	Specifies when the DNAT rule is created (UTC time). Its value rounds to 6 decimal places for seconds. The format is yyyy-mm-dd hh:mm:ss.

Examples

- Example request
  ```
  GET https://{Endpoint}/v2.0/dnat_rules/79195d50-0271-41f1-bded-4c089b2502ff
  ```
- Example response
  ```
  {
      "dnat_rule": {
        "floating_ip_id": "bf99c679-9f41-4dac-8513-9c9228e713e1",
         "status": "ACTIVE",
         "nat_gateway_id": "cda3a125-2406-456c-a11f-598e10578541",
        "admin_state_up": true,
         "port_id": "9a469561-daac-4c94-88f5-39366e5ea193",
         "internal_service_port": 993,
         "protocol": "TCP",
         "tenant_id": "abc",
        "created_at": "2017-11-15 15:44:42.595173",
         "id": "79195d50-0271-41f1-bded-4c089b2502ff",
        "floating_ip_address": "5.21.11.226",
        "external_service_port": 242
        "private_ip": ""
    }
  }
  ```

Status Codes

See **Status Codes**.

8.1.3.4 Deleting a DNAT Rule

Function

This API is used to delete a DNAT rule.

URI

DELETE /v2.0/dnat_rules/{dnat_rule_id}

Table 8-38 Parameter description

Parameter	Mandatory	Type	Description
dnat_rule_id	Yes	String	Specifies the ID of the DNAT rule.

Request

None

Response

None

Examples

- Example request


```
DELETE https://{Endpoint}/v2.0/dnat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
```

- Example response

```
None (STATUS CODE 204)
```

Status Code

See **Status Codes**.

8.1.4 Tags

8.1.4.1 Querying NAT Gateways by Tag

Function

This API is used to filter NAT gateways by tag.

TMS uses this API to filter and list NAT gateways by tag.

By default, NAT gateways and tags are sorted in descending order of creation time.

URI

- URI format

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

- Parameter description

Table 8-39 Parameter description

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

Request

Table 8-40 describes the request parameters.

Table 8-40 Request parameters

Parameter	Mandatory	Type	Description
tags	No	Array<Object>	Specifies the included tags. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Each tag key must be unique, and each tag value in a tag must be unique. Resources identified by different keys are in AND relationship, and values in one tag are in OR relationship. If no tag filtering criteria is specified, full data is returned.
tags_any	No	Array<Object>	Specifies any included tags. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Each tag key must be unique, and each tag value in a tag must be unique. Resources identified by different keys are in OR relationship, and values in one tag are in OR relationship. If no tag filtering criteria is specified, full data is returned.
not_tags	No	Array<Object>	Specifies the excluded tags. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Each tag key must be unique, and each tag value in a tag must be unique. Resources not identified by different keys are in AND relationship, and values in one tag are in OR relationship. If no tag filtering criteria is specified, full data is returned.

Parameter	Mandatory	Type	Description
not_tags_any	No	Array<Object>	Specifies any excluded tags. Each tag contains a maximum of 10 keys, and each key contains a maximum of 10 values. The structure body cannot be missing, and the key cannot be left blank or set to an empty string. Each tag key must be unique, and each tag value in a tag must be unique. Resources not identified by different keys are in OR relationship, and values in one tag are in OR relationship. If no tag filtering criteria is specified, full data is returned.
limit	No	String	Number of records. 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	No	String	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	Yes	String	Specifies the operation to perform, which can only be **filter** (filtering) or **count** (querying the total number). The value **filter** indicates pagination query. Value **count** indicates that the total number of query results meeting the search criteria will be returned. Returning other fields is not allowed.

Parameter	Mandatory	Type	Description
matches	No	Array<match>	Specifies the search criteria. The tag key is the field to match, for example, **resource_name**. The tag value indicates the matched value. This field is a fixed dictionary value. Determine whether fuzzy match is required based on different fields. For example, if **key** is **resource_name**, fuzzy search (case insensitive) is used by default. If **value** is an empty string, exact match is used. If **key** is **resource_id**, exact match is used.

Table 8-41 Parameter description of field **tag**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It can contain a maximum of 127 Unicode characters. **key** cannot be left blank, be an empty string, or be spaces. Before using key, delete spaces before and after the value.
values	Yes	Array<String>	Specifies the tag values. Each value contains a maximum of 255 Unicode characters and cannot contain spaces. Before verification, delete spaces before and after the value. The asterisk (*) is a reserved character. The value can be empty but cannot be left blank. If the value starts with an asterisk (*), the string following the asterisk is fuzzy matched. If the values are null, it indicates **any_value** (querying any value). The values are in OR relationship.

Table 8-42 Description of field **match**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. The value is fixed at **resource_name**.
value	Yes	String	Specifies the key value. It can contain a maximum of 255 Unicode characters.

Response

Table 8-43 describes the response parameters.

Table 8-43 Response parameter

Parameter	Type	Description
resources	Array<resource>	Specifies the resource object list. For details, see **Table 8-44**.
total_count	Integer	Specifies the total number of resources.

Table 8-44 Data structure description of field **resource**

Parameter	Type	Description
resource_id	String	Specifies the resource ID.
resource_detail	Object	Provides details about the resource. The value is a resource object, used for extension. This parameter is left blank by default.
tags	Array<resource_tag>	Specifies the list of queried tags. If no tag is matched, an empty array is returned. For details, see **Table 8-45**.
resource_name	String	Specifies the resource name. This parameter is an empty string by default if there is no resource name.

Table 8-45 Parameter description of field **resource_tag**

Parameter	Type	Description
key	String	Specifies the tag key. It contains a maximum of 36 Unicode characters. A tag key cannot be left blank. It cannot contain non-printable ASCII characters (0–31) or the following special characters: *<>\=
value	String	Specifies the key value. The value can contain a maximum of 43 Unicode characters and can be an empty string. It cannot contain ASCII characters (0–31) and the following characters: *<>\=

Examples

- Example request

  a. Request body when **action** is set to **filter**

```
POST  https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/
resource_instances/action
{
   "offset": "100",
   "limit": "100",
   "action": "filter",
   "matches": [
      {
         "key": "resource_name",
         "value": "nat_gateways"
      }
   ],
   "not_tags": [
      {
         "key": "key1",
         "values": [
            "*value1",
            "value2"
         ]
      }
   ],
   "tags": [
      {
         "key": "key2",
         "values": [
            "*value3",
            "value4"
         ]
      }
   ],
   "tags_any": [
      {
         "key": "key3",
         "values": [
```



```
            "*value5",
            "value6"
         ]
      }
   ],
   "not_tags_any": [
      {
         "key": "key4",
         "values": [
            "*value7",
            "value8"
         ]
      }
   ]
}
```

b. Request body when **action** is set to **count**

```
POST  https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/
resource_instances/action        {
   "action": "count",
   "matches": [
      {
         "key": "resource_name",
         "value": "nat_gateways"
      }
   ],
   "not_tags": [
      {
         "key": "key1",
         "values": [
            "*value1",
            "value2"
         ]
      }
   ],
   "tags": [
      {
         "key": "key2",
         "values": [
            "*value3",
            "value4"
         ]
      }
   ],
   "tags_any": [
      {
         "key": "key3",
         "values": [
            "*value5",
            "value6"
         ]
      }
   ],
   "not_tags_any": [
      {
         "key": "key4",
         "values": [
            "*value7",
            "value8"
         ]
      }
   ]
}
```

- Example response

a. Response body when **action** is set to **filter**

```
{
   "resources": [
```



```
    {
      "resource_detail": null,
      "resource_id": "e5ad289f-9c56-4daf-b08b-2e53a983473a",
      "resource_name": "nat_gateways",
      "tags": [
        {
          "key": "key2",
          "value": "value4"
        },
        {
          "key": "key2",
          "value": "value3"
        }
      ]
    }
  ],
  "total_count": 1000
}
```


b. Response body when **action** is set to **count**


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


Status Codes

See **Status Codes**.

8.1.4.2 Adding or Deleting NAT Gateway Tags in Batches

Function

This API is used to add or delete tags of a specific NAT gateway in batches.

TMS uses this API to manage resource tags.

You can add a maximum of 10 tags to a NAT gateway.

This API is idempotent.

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

If a to-be-created tag has the same key as an existing tag, the tag will be created and overwrite the existing one.

When tags are being deleted and some tags do not exist, the operation is considered to be successful by default. The character set of the tags will not be checked. A key and a value can respectively contain up to 127 and 255 Unicode characters. When you delete tags, the tag structure cannot be missing, and the key cannot be left blank or be an empty string.

URI

- URI format

POST /v2.0/{project_id}/nat_gateways/{nat_gateway_id}/tags/action

Table 8-46 Parameter description

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

Request

Table 8-47 describes the request parameters.

Table 8-47 Request parameter

Parameter	Mandatory	Type	Description
tags	Yes	Array	Specifies tags. For details, see **Table 8-48**.
action	Yes	String	Specifies the operation to be performed, which can be set to **create** or **delete** only.

Table 8-48 Parameter description of field **tags**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It contains a maximum of 36 Unicode characters. A tag key cannot be left blank. It cannot contain ASCII (0-31) or the following characters: =*<>\,|/
value	Yes	String	Specifies the key value. Each value contains a maximum of 43 Unicode characters. If **value** is specified, tags are deleted by key and value. If **value** is not specified, tags are deleted by key. The value can be an empty character string. It cannot contain ASCII (0-31) or the following characters: =*<>\,|/

Response

None

Examples

- Example request

```
POST  https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/
fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags/action
{
   "action": "create",
   "tags": [
      {
         "key": "key1",
         "value": "value1"
      },
      {
         "key": "key2",
         "value": "value2"
      }
   ]
}
Or
{
   "action": "delete",
   "tags": [
      {
         "key": "key1",
         "value": "value1
       },
      {
         "key": "key2",
         "value": "value2"
      }
   ]
}
```

Example response

None

Status Codes

See **Status Codes**.

8.1.4.3 Adding a NAT Gateway Tag

Function

This API is used to add tags to a NAT gateway. You can add a maximum of 10 tags to a NAT gateway.

This API is idempotent.

If a to-be-created tag has the same key as an existing tag, the tag will be created and overwrite the existing one.

NOTE

Ensure that the NAT gateway to which you will add tags is available.

URI

POST /v2.0/{project_id}/nat_gateways/{nat_gateway_id}/tags

Table 8-49 Parameter description

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

Request

Table 8-50 describes the request parameters.

Table 8-50 Request parameter

Parameter	Mandatory	Type	Description
tag	Yes	Array	Specifies the list of tags.

Table 8-51 Parameter description of field **tag**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It contains a maximum of 36 Unicode characters. A tag key cannot be left blank. It cannot contain ASCII (0-31) or the following characters: =*<>\,|/
value	Yes	String	Specifies the key value. The value can contain a maximum of 43 Unicode characters and can be an empty string. It cannot contain ASCII (0-31) or the following characters: =*<>\,|/

Response

None

Examples

- Example request

```
POST  https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags
{
    "tag":
   {
```



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

Status Codes

See **Status Codes**.

8.1.4.4 Deleting a NAT Gateway Tag

Function

This API is idempotent.

When a tag is deleted, it is not verified. The tag key cannot be left blank or be an empty string. If the key of the tag to be deleted is not found, 404 will be returned.

URI

DELETE /v2.0/{project_id}/nat_gateways/{nat_gateway_id}/tags/{key}

Table 8-52 Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Specifies the project ID.
nat_gateway_id	Yes	String	Specifies the NAT gateway ID.
key	Yes	String	Specifies the tag key.

Request

None

Response

None

Examples

- Example request
  ```
  DELETE https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags/key1
  ```
- Example response
  ```
  None (STATUS CODE 204)
  ```

Status Codes

See **Status Codes**.

8.1.4.5 Querying NAT Gateway Tags

Function

This API is used to query tags of a specified NAT gateway.

TMS uses this API to query all tags of a specified NAT gateway.

URI

GET /v2.0/{project_id}/nat_gateways/{nat_gateway_id}/tags

Table 8-53 Parameter description

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

Request

None

Response

Table 8-54 describes the response parameters.

Table 8-54 Response parameter

Parameter	Mandatory	Type	Description
tags	Yes	Array	Specifies the list of tags.

Table 8-55 Parameter description of field **tags**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It contains a maximum of 36 Unicode characters. A tag key cannot be left blank. It cannot contain non-printable ASCII characters (0–31) or the following special characters: *<>\=

Parameter	Mandatory	Type	Description
value	Yes	String	Specifies the key value. The value can contain a maximum of 43 Unicode characters and can be an empty string. It cannot contain non-printable ASCII characters (0–31) or the following special characters: *<>\=

Examples

- Example request
  ```
  GET https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags
  ```
- Example response
  ```
  {
     "tags": [
        {
           "key": "key1",
           "value": "value1"
        },
        {
           "key": "key2",
           "value": "value2"
        }
     ]
  }
  ```

Status Codes

See **Status Codes**.

8.1.4.6 Querying Tags in a Project

Function

This API is used to query all tags of a resource type in a specified region.

TMS uses this API to list tags created by a tenant to facilitate tag creation and resource filtering on the console.

URI

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

Table 8-56 Parameter description

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

Request

None

Response

Table 8-57 describes the response parameters.

Table 8-57 Response parameter

Parameter	Mandatory	Type	Description
tags	Yes	Array	Specifies the list of tags.

Table 8-58 Parameter description of field **tags**

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. It contains a maximum of 36 Unicode characters. A tag key cannot be left blank. It cannot contain non-printable ASCII characters (0–31) or the following special characters: *<>\=
values	Yes	Array<String>	Specifies the tag values. The value can contain a maximum of 43 Unicode characters and can be an empty string. It cannot contain non-printable ASCII characters (0–31) or the following special characters: *<>\=

Examples

- Example request
  ```
  GET https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/tags
  ```
- Example response
  ```
  {
    "tags": [
  ```



```
    {
        "key": "key1",
        "values": [
            "value1",
            "value2"
        ]
    },
    {
        "key": "key2",
        "values": [
            "value3",
            "value4"
        ]
    }
  ]
}
```

Status Codes

See **Status Codes**.

A Change History

Released On	Description
2022-10-30	This issue is the second official issue, which incorporates the following change: Added **Application Examples**.
2022-04-12	This issue is the first official release.

