NAT 网关

API 参考

文档版本 11

发布日期 2024-03-27

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 使用前必读

1.1 概述

欢迎使用NAT网关（NAT Gateway）。NAT网关能够为虚拟私有云内的云主机（弹性云服务器、裸金属服务器）或者通过云专线/VPN接入虚拟私有云的本地数据中心的服务器，提供网络地址转换服务，使多个云主机可以共享弹性公网IP访问Internet或使云主机提供互联网服务。

您可以使用本文档提供的API对NAT网关进行相关操作，如创建、删除、添加SNAT规则等。支持的全部操作请参考**API概览**。

在调用NAT网关API之前，请确保已经充分了解NAT网关相关概念，详细信息请参考**产品介绍**。

1.2 调用说明

NAT网关提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参考**如何调用API**。

1.3 终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从**地区和终端节点**中查询NAT网关的终端节点。

1.4 约束与限制

- NAT网关使用过程中的约束与限制，具体请参见**约束与限制**。
- 更详细的限制请参见具体API的说明。

1.5 基本概念

- 账号

  用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建

议您不要直接使用账号进行日常管理工作，而是创建用户并使用用户进行日常管理工作。

- 用户

  由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

  通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

  从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

  详情请参见**区域和可用区**。

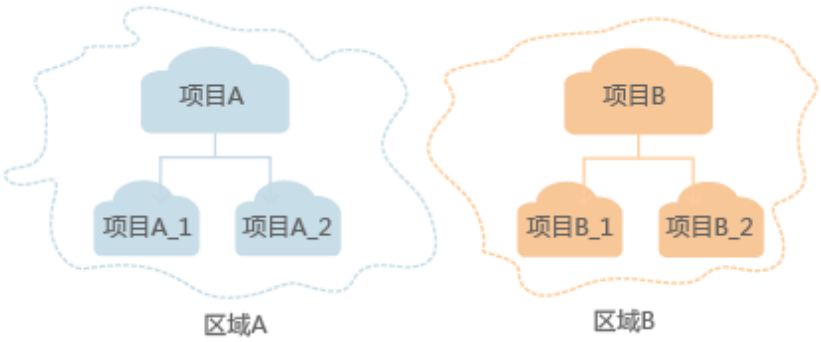
- 可用区（AZ，Availability Zone）

  一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 项目

  区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

  图 1-1 项目隔离模型



- 企业项目

  企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

  关于企业项目ID的获取及企业项目特性的详细信息，请参见《**企业管理用户指南**》。

1.6 API 版本选择建议

NAT网关支持v2.0和v2两个版本的API接口。v2版本接口为较新的版本，产品将持续对v2版本的接口进行优化和功能加强，推荐您使用v2版本API接口。

2 API 概览

通过使用NAT网关服务所提供的接口，您可以完整的使用NAT网关服务的所有功能。

公网 NAT 网关

表 2-1 公网 NAT 网关接口类型

类型	说明
公网NAT网关	支持对公网NAT网关进行创建、查询、更新、删除等操作，包括创建公网NAT网关、查询公网NAT网关列表、查询指定的公网NAT网关详情、更新公网NAT网关、删除公网NAT网关等接口。
SNAT规则	支持创建、查询、更新、删除SNAT规则，包括创建SNAT规则、查询SNAT规则列表、查询指定的SNAT规则详情、更新SNAT规则、删除SNAT规则等接口。
DNAT规则	支持创建、查询、更新、删除DNAT规则，包括创建DNAT规则、批量创建DNAT规则、查询DNAT规则列表、查询指定的DNAT规则详情、更新DNAT规则、删除DNAT规则等接口。
公网NAT网关标签	公网NAT网关的标签可以对公网NAT网关进行自定义标记，通过标签，用户可以对公网NAT网关进行管理，包括查询、添加、删除标签等操作。

表 2-2 公网 NAT 网关接口说明

类型	API	说明
公网NAT网关	创建公网NAT网关	创建公网NAT网关实例。
	查询公网NAT网关列表	查询公网NAT网关列表。
	查询指定的公网NAT网关详情	查询指定的公网NAT网关详情。
	更新公网NAT网关	更新公网NAT网关。

类型	API	说明
	删除公网NAT网关	删除公网NAT网关。
DNAT规则	创建DNAT规则	创建DNAT规则。
	批量创建DNAT规则	批量创建DNAT规则。
	查询DNAT规则列表	查询DNAT规则列表。
	查询指定的DNAT规则详情	查询指定的DNAT规则详情。
	更新DNAT规则	更新DNAT规则。
	删除DNAT规则	删除DNAT规则。
SNAT规则	创建SNAT规则	创建SNAT规则。
	查询SNAT规则列表	查询SNAT规则列表。
	查询指定的SNAT规则详情	查询指定的SNAT规则详情。
	更新SNAT规则	更新SNAT规则。
	删除SNAT规则	删除SNAT规则。
公网NAT网关标签	查询公网NAT网关资源实例	使用标签过滤公网NAT网关资源实例。
	批量添加/删除公网NAT网关资源标签	为指定公网NAT网关实例批量添加或删除标签。
	添加公网NAT网关资源标签	添加公网NAT网关资源标签。
	删除公网NAT网关资源标签	删除公网NAT网关资源标签。
	查询公网NAT网关资源标签	查询指定公网NAT网关实例的标签信息。
	查询公网NAT网关项目标签	查询租户在指定Region和公网NAT网关实例类型的所有标签集合。

私网 NAT 网关

表 2-3 私网 NAT 网关接口类型

类型	说明
私网NAT网关	支持对私网NAT网关进行查询、更新、删除、创建等操作，包括查询私网NAT网关列表、更新私网NAT网关、删除私网NAT网关、创建私网NAT网关、查询指定的私网NAT网关详情接口。

类型	说明
DNAT规则	支持查询、更新、创建、删除DNAT规则，包括查询DNAT规则列表、更新DNAT规则、创建DNAT规则、删除DNAT规则、查询指定的DNAT规则详情接口。
SNAT规则	支持查询、更新、创建、删除SNAT规则，包括查询SNAT规则列表、查询指定的SNAT规则详情、更新SNAT规则、创建SNAT规则、删除SNAT规则接口。
中转IP	支持查询、删除、创建中转IP，包括查询中转IP列表、删除中转IP、创建中转IP、查询指定的中转IP详情接口。
私网NAT网关标签管理	私网NAT网关的标签可以对NAT网关进行自定义标记，通过标签，用户可以对私网NAT网关进行管理，包括查询、添加、删除标签等操作。
中转IP标签管理	中转IP的标签可以对中转IP进行自定义标记，通过标签，用户可以对中转IP进行管理，包括查询、添加、删除标签等操作。

表 2-4 私网 NAT 网关接口说明

类型	API	说明
私网NAT网关	查询私网NAT网关列表	查询私网NAT网关列表。
	更新私网NAT网关	更新私网NAT网关。
	删除私网NAT网关	删除私网NAT网关。
	创建私网NAT网关	创建私网NAT网关实例。
	查询指定的私网NAT网关详情	查询指定的私网NAT网关详情。
DNAT规则	查询DNAT规则列表	查询DNAT规则列表。
	更新DNAT规则	更新DNAT规则。
	创建DNAT规则	创建DNAT规则。
	删除DNAT规则	删除DNAT规则。
	查询指定的DNAT规则详情	查询指定的DNAT规则详情。
SNAT规则	查询SNAT规则列表	查询SNAT规则列表。
	查询指定的SNAT规则详情	查询指定的SNAT规则详情。
	更新SNAT规则	更新SNAT规则。
	创建SNAT规则	创建SNAT规则。
	删除SNAT规则	删除SNAT规则。

类型	API	说明
中转IP	查询中转IP列表	查询中转IP列表。
	删除中转IP	删除中转IP。
	创建中转IP	创建中转IP。
	查询指定的中转IP详情	查询指定的中转IP详情。
私网NAT网关标签管理	查询私网NAT网关实例	使用标签过滤私网NAT网关实例。
	查询私网NAT网关项目标签	查询租户在指定Project的所有私网NAT网关标签集合。
	查询私网NAT网关标签	查询指定私网NAT网关实例的标签信息。
	添加私网NAT网关标签	添加私网NAT网关资源标签。
	批量添加删除私网NAT网关标签	为指定私网NAT网关实例批量添加或删除标签。
	删除私网NAT网关标签	删除私网NAT网关资源标签。
中转IP标签管理	查询中转IP实例	使用标签过滤中转IP实例。
	查询中转IP项目标签	查询租户在指定Project的所有中转IP标签集合。
	查询中转IP标签	查询指定中转IP实例的标签信息。
	添加中转IP标签	添加中转IP标签
	批量添加删除中转IP标签	为指定中转IP实例批量添加或删除标签
	删除中转IP标签	删除中转IP标签

API v2.0

表 2-5 NAT 网关 v2.0 接口类型

类型	说明
公网NAT网关	支持对公网NAT网关进行创建、查询、更新、删除等操作，包括创建公网NAT网关、查询公网NAT网关列表、查询公网NAT网关、更新公网NAT网关、删除公网NAT网关等接口。
SNAT规则	支持创建、查询、删除SNAT规则，包括创建SNAT规则、查询SNAT规则列表、查询SNAT规则、删除SNAT规则等接口。
DNAT规则	支持创建、查询、删除DNAT规则，包括创建DNAT规则、查询DNAT规则列表、查询DNAT规则、删除DNAT规则等接口。

表 2-6 NAT 网关 v2.0 接口说明

类型	API	说明
公网NAT网关	创建公网NAT网关	创建公网NAT网关实例。
	查询公网NAT网关列表	查询公网NAT网关列表。
	查询指定的公网NAT网关详情	查询指定的公网NAT网关详情。
	更新公网NAT网关	更新公网NAT网关。
	删除公网NAT网关	删除公网NAT网关。
SNAT规则	创建SNAT规则	创建SNAT规则。
	查询SNAT规则列表	查询SNAT规则列表。
	查询指定的SNAT规则详情	查询指定的SNAT规则详情。
	删除SNAT规则	删除SNAT规则。
DNAT规则	创建DNAT规则	创建DNAT规则
	查询DNAT规则列表	查询DNAT规则列表。
	查询指定的DNAT规则详情	查询指定的DNAT规则详情。
	删除DNAT规则	删除DNAT规则。

3 如何调用 API

3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的**获取用户Token**说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

请求 URI

请求URI由如下部分组成：

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

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

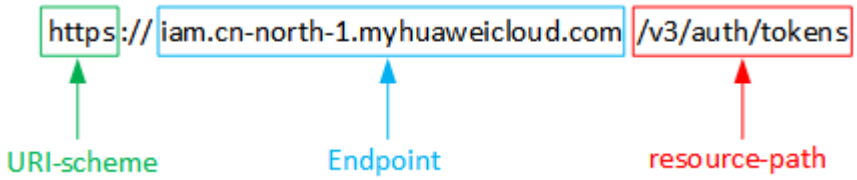
表 3-1 URI 中的参数说明

参数	描述
URI-scheme	表示用于传输请求的协议，当前所有API均采用**HTTPS**协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从**地区和终端节点**获取。 例如IAM服务在“华北-北京一”区域的Endpoint为“iam.cn-north-1.myhuaweicloud.com”。
resource-path	资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如您需要获取IAM在“华北-北京一”区域的Token，则需使用“华北-北京一”区域的Endpoint（iam.cn-north-1.myhuaweicloud.com），并在**获取用户Token**的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

图 3-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

表 3-2 HTTP 方法

方法	说明
GET	请求服务器返回指定资源。
PUT	请求服务器更新指定资源。
POST	请求服务器新增资源或执行特殊操作。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	请求服务器资源头部。
PATCH	请求服务器更新资源的部分内容。 当资源不存在的时候，PATCH可能会去创建一个新的资源。

在**获取用户Token**的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

详细的公共请求消息头字段请参见**表3-3**。

表 3-3 公共请求消息头

名称	描述	是否必选	示例
Host	请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。	否 使用AK/SK认证时该字段必选。	code.test.com or code.test.com:443
Content-Type	消息体的类型（格式）。推荐用户使用默认值application/json，有其他取值时会在具体接口中专门说明。	是	application/json
Content-Length	请求body长度，单位为Byte。	否	3495
X-Project-Id	project id，项目编号。请参考**获取项目ID**章节获取项目编号。	否 如果是专属云场景采用AK/SK认证方式的接口请求或者多project场景采用AK/SK认证的接口请求，则该字段必选。	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	用户Token。 用户Token也就是调用**获取用户Token**接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头（Headers）中包含的“X-Subject-Token”的值即为Token值。	否 使用Token认证时该字段必选。	注：以下仅为Token示例片段 MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

说明

API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

AK/SK认证的详细说明请参见**认证鉴权**的“AK/SK认证”。

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-Type”即可，添加消息头后的请求如下所示。

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

请求消息体（可选）

该部分可选。请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-Type对应，传递除请求消息头之外的内容。若请求消息体中的参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于**获取用户Token**接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写，其中***username***为用户名，***domainname***为用户所属的账号名称，*********为用户登录密码，***xxxxxxxxxxxxxxxxx***为project的名称，如“cn-north-1”，您可以从**地区和终端节点**获取。

说明

scope参数定义了Token的作用域，下面示例中获取的Token仅能访问project下的资源。您还可以设置Token的作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见**获取用户Token**。

```
POST https://iam.cn-north-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"
            }
        }
    }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用**curl**、**Postman**或直接编写代码等方式发送请求调用API。对于获取用户Token接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证通用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

Token 认证

说明

Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用**获取用户Token**接口获取，调用本服务API需要project级别的Token，即调用**获取用户Token**接口时，请求body中auth.scope的取值需要选择project，如下所示。

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

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

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

AK/SK 认证

说明

AK/SK签名认证方式仅支持消息体大小12MB以内，12MB以上的请求请使用Token认证。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK（Access Key ID）：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK（Secret Access Key）：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见**API签名指南**。

说明

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

3.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见**状态码**。

对于**获取用户Token**接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于**获取用户Token**接口，返回如**图3-2**所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 3-2 获取用户 Token 响应消息头

响应消息体（可选）

该部分可选。响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-Type对应，传递除响应消息头之外的内容。

对于**获取用户Token**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

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

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

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

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API（公网 NAT 网关）

4.1 公网 NAT 网关

4.1.1 查询公网 NAT 网关列表

功能介绍

查询公网NAT网关实例列表。

接口约束

可以在URI后面用'?'和'&'添加不同的查询条件组合。支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/nat_gateways

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 4-2 Query 参数

参数	是否必选	参数类型	描述
id	否	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
enterprise_project_id	否	String	企业项目ID。创建公网NAT网关实例时，关联的企业项目ID。 缺省值：**0** 最小长度：**1** 最大长度：**36**
description	否	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
created_at	否	String	公网NAT网关实例的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
name	否	String	公网NAT网关实例的名字，长度限制为64。 公网NAT网关实例的名字仅支持数字、字母、_（下划线）、-（中划线）、中文 最小长度：**1** 最大长度：**64**
status	否	Array	公网NAT网关实例的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "INACTIVE"：不可用 数组长度：**1 - 10** 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **INACTIVE**

参数	是否必选	参数类型	描述
spec	否	Array	公网NAT网关实例的规格。 取值为： "1"：小型，SNAT最大连接数10000 "2"：中型，SNAT最大连接数50000 "3"：大型，SNAT最大连接数200000 "4"：超大型，SNAT最大连接数1000000 数组长度：**1 - 10** 枚举值： • **1** • **2** • **3** • **4**
admin_state_up	否	Boolean	解冻/冻结状态。 取值范围： "true"：解冻 "false"：冻结
internal_network_id	否	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
router_id	否	String	VPC的id。 最小长度：**36** 最大长度：**36**
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**

参数	是否必选	参数类型	描述
marker	否	String	分页查询的起始资源ID，表示从指定资源的下一条记录开始查询。 ● 若不传入marker和limit参数，查询结果返回第一页全部资源记录（默认2000条）。 ● 若不传入marker参数，limit为10，查询结果返回第1~10条资源记录。 ● 若marker为第10条记录的资源ID，limit为10，查询结果返回第11~20条资源记录。 ● 若marker为第10条记录的资源ID，不传入limit参数，查询结果返回第11条及之后的资源记录（默认2000条）。 最小值：**36** 最大值：**36**

请求参数

表 4-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-4 响应 Body 参数

参数	参数类型	描述
nat_gateways	Array of **NatGatewayResponseBody** objects	查询公网NAT网关实例列表的响应体。 详见NatGateway字段说明。 数组长度：**2000 - 0**

表 4-5 NatGatewayResponseBody

参数	参数类型	描述
id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
name	String	公网NAT网关实例的名字，长度限制为64。 最小长度：**1** 最大长度：**64**
description	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	公网NAT网关的规格。 取值为： “1”：小型，SNAT最大连接数10000 “2”：中型，SNAT最大连接数50000 “3”：大型，SNAT最大连接数200000 “4”：超大型，SNAT最大连接数1000000 枚举值： • **1** • **2** • **3** • **4**

参数	参数类型	描述
status	String	公网NAT网关实例的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE": 创建中 "PENDING_UPDATE": 更新中 "PENDING_DELETE": 删除中 "INACTIVE": 不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： • "true": 解冻 • "false": 冻结
created_at	String	公网NAT网关实例的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
router_id	String	VPC的id。 最小长度：**36** 最大长度：**36**
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。 创建公网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
session_conf	**SessionConfiguration** object	公网NAT网关会话参数配置。
ngport_ip_address	String	公网NAT网关私有IP地址，由VPC中子网分配。 最小长度：**7** 最大长度：**15**
billing_info	String	订单信息。此字段只有在订购包周期资源时才会有订单信息，而在订购按需资源时则为空。

参数	参数类型	描述
dnat_rules_limit	Long	公网NAT网关下DNAT规则数量限制，默认为200。 最小值：**1** 最大值：**100000**
snat_rule_public_ip_limit	Integer	公网NAT网关下SNAT规则EIP池中EIP数量限制，默认为20。 最小值：**1** 最大值：**100**

表 4-6 SessionConfiguration

参数	参数类型	描述
tcp_session_expire_time	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**
tcp_time_wait_time	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

请求示例

```
GET https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/nat_gateways?status=ACTIVE
```

响应示例

状态码： 200

查询公网NAT网关实例列表成功

```
{
  "nat_gateways" : [ {
    "id" : "a253be25-ae7c-4013-978b-3c0785eccd63",
    "router_id" : "b1d81744-5165-48b8-916e-e56626feb88f",
    "status" : "ACTIVE",
    "description" : "nat01",
    "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",
  "name" : "wj3",
  "enterprise_project_id" : "0aad99bc-f5f6-4f78-8404-c598d76b0ed2",
  "billing_info" : "",
  "dnat_rules_limit" : 200,
  "snat_rule_public_ip_limit" : 20,
  "ngport_ip_address" : "192.168.0.138"
 }, {
  "id" : "e824f1b4-4290-4ebc-8322-cfff370dbd1e",
  "router_id" : "305dc52f-13dd-429b-a2d4-444a1039ba0b",
  "status" : "ACTIVE",
  "description" : "1234",
  "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",
  "name" : "lyl001",
  "enterprise_project_id" : "0",
  "billing_info" : "",
  "dnat_rules_limit" : 200,
  "snat_rule_public_ip_limit" : 20,
  "ngport_ip_address" : "192.168.5.210"
 } ]
}
```


状态码

状态码	描述
200	查询公网NAT网关实例列表成功

错误码

请参见**错误码**。

4.1.2 创建公网 NAT 网关

功能介绍

创建公网NAT网关实例。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/nat_gateways

表 4-7 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 4-8 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-9 请求 Body 参数

参数	是否必选	参数类型	描述
nat_gateway	是	**CreateNatGatewayOption** object	创建公网NAT网关实例的请求体。

表 4-10 CreateNatGatewayOption

参数	是否必选	参数类型	描述
name	是	String	公网NAT网关实例的名字，长度限制为64。 公网NAT网关实例的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。 最小长度：**1** 最大长度：**64**

参数	是否必选	参数类型	描述
router_id	是	String	VPC的id。 最小长度：**36** 最大长度：**36**
internal_network_id	是	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
description	否	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	是	String	公网NAT网关的规格。 取值为： “1”：小型，SNAT最大连接数10000 “2”：中型，SNAT最大连接数50000 “3”：大型，SNAT最大连接数200000 “4”：超大型，SNAT最大连接数1000000 枚举值： • **1** • **2** • **3** • **4**
enterprise_project_id	否	String	企业项目ID。 创建公网NAT网关实例时，关联的企业项目ID。关于企业项目ID的获取及企业项目特性的详细信息，请参考《企业管理用户指南》。 缺省值：**0** 最小长度：**1** 最大长度：**36**
session_conf	否	**SessionConfiguration** object	公网NAT网关会话参数配置。
ngport_ip_address	否	String	公网NAT网关私有IP地址，由VPC中子网分配。 最小长度：**7** 最大长度：**15**

表 4-11 SessionConfiguration

参数	是否必选	参数类型	描述
tcp_session_expire_time	否	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	否	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	否	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**
tcp_time_wait_time	否	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

响应参数

状态码： 201

表 4-12 响应 Body 参数

参数	参数类型	描述
nat_gateway	**NatGatewayResponseBody** object	公网NAT网关实例的响应体。

表 4-13 NatGatewayResponseBody

参数	参数类型	描述
id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**

参数	参数类型	描述
name	String	公网NAT网关实例的名字，长度限制为64。 最小长度：**1** 最大长度：**64**
description	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	公网NAT网关的规格。 取值为： “1”：小型，SNAT最大连接数10000 “2”：中型，SNAT最大连接数50000 “3”：大型，SNAT最大连接数200000 “4”：超大型，SNAT最大连接数1000000 枚举值： ● **1** ● **2** ● **3** ● **4**
status	String	公网NAT网关实例的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "INACTIVE"：不可用 枚举值： ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： ● "true"：解冻 ● "false"：冻结
created_at	String	公网NAT网关实例的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
router_id	String	VPC的id。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。 创建公网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
session_conf	**SessionConfiguration** object	公网NAT网关会话参数配置。
ngport_ip_address	String	公网NAT网关私有IP地址，由VPC中子网分配。 最小长度：**7** 最大长度：**15**
billing_info	String	订单信息。此字段只有在订购包周期资源时才会有订单信息，而在订购按需资源时则为空。
dnat_rules_limit	Long	公网NAT网关下DNAT规则数量限制，默认为200。 最小值：**1** 最大值：**100000**
snat_rule_public_ip_limit	Integer	公网NAT网关下SNAT规则EIP池中EIP数量限制，默认为20。 最小值：**1** 最大值：**100**

表 4-14 SessionConfiguration

参数	参数类型	描述
tcp_session_expire_time	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**

参数	参数类型	描述
tcp_time_wait_time	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

请求示例

创建公网NAT网关

```
POST https://{Endpoint}/v2/70505c941b9b4dfd82fd351932328a2f/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"
  }
}
```

响应示例

状态码： 201

创建公网NAT网关实例成功。

```
{
  "nat_gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "nat_001",
    "description" : "my nat gateway 01",
    "router_id" : "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
    "spec" : "1",
    "admin_state_up" : true,
    "tenant_id" : "70505c941b9b4dfd82fd351932328a2f",
    "internal_network_id" : "89d66639-aacb-4929-969d-07080b0f9fd9",
    "enterprise_project_id" : "0aad99bc-f5f6-4f78-8404-c598d76b0ed2",
    "status" : "PENDING_CREATE",
    "billing_info" : "",
    "dnat_rules_limit" : 200,
    "snat_rule_public_ip_limit" : 20,
    "ngport_ip_address" : "192.168.0.138",
    "created_at" : "2019-04-22 08:47:13.234512"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建公网NAT网关

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreateNatGatewaySolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateNatGatewayRequest request = new CreateNatGatewayRequest();
        CreateNatGatewayRequestBody body = new CreateNatGatewayRequestBody();
        CreateNatGatewayOption natGatewaybody = new CreateNatGatewayOption();
        natGatewaybody.withName("nat_001")
            .withRouterId("d84f345c-80a1-4fa2-a39c-d0d397c3f09a")
            .withInternalNetworkId("89d66639-aacb-4929-969d-07080b0f9fd9")
            .withDescription("my nat gateway 01")
            .withSpec(CreateNatGatewayOption.SpecEnum.fromValue("1"))
            .withEnterpriseProjectId("0aad99bc-f5f6-4f78-8404-c598d76b0ed2");
        body.withNatGateway(natGatewaybody);
        request.withBody(body);
        try {
            CreateNatGatewayResponse response = client.createNatGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建公网NAT网关

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateNatGatewayRequest()
        natGatewaybody = CreateNatGatewayOption(
            name="nat_001",
            router_id="d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
            internal_network_id="89d66639-aacb-4929-969d-07080b0f9fd9",
            description="my nat gateway 01",
            spec="1",
            enterprise_project_id="0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
        )
        request.body = CreateNatGatewayRequestBody(
            nat_gateway=natGatewaybody
        )
        response = client.create_nat_gateway(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建公网NAT网关

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.CreateNatGatewayRequest{}
    descriptionNatGateway:= "my nat gateway 01"
    enterpriseProjectIdNatGateway:= "0aad99bc-f5f6-4f78-8404-c598d76b0ed2"
    natGatewaybody := &model.CreateNatGatewayOption{
        Name: "nat_001",
        RouterId: "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
        InternalNetworkId: "89d66639-aacb-4929-969d-07080b0f9fd9",
        Description: &descriptionNatGateway,
        Spec: model.GetCreateNatGatewayOptionSpecEnum().E_1,
        EnterpriseProjectId: &enterpriseProjectIdNatGateway,
    }
    request.Body = &model.CreateNatGatewayRequestBody{
        NatGateway: natGatewaybody,
    }
    response, err := client.CreateNatGateway(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建公网NAT网关实例成功。

错误码

请参见**错误码**。

4.1.3 删除公网 NAT 网关

功能介绍

删除公网NAT网关实例。

调用方法

请参见**如何调用API**。

URI

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

表 4-15 路径参数

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 4-16 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{NAT_endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/nat_gateways/
a78fb3eb-1654-4710-8742-3fc49d5f04f8
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeleteNatGatewaySolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteNatGatewayRequest request = new DeleteNatGatewayRequest();
        try {
            DeleteNatGatewayResponse response = client.deleteNatGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

    client = NatClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(NatRegion.value_of("<YOUR REGION>")) \
```

```
        .build()

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除公网NAT网关实例成功。

错误码

请参见**错误码**。

4.1.4 更新公网 NAT 网关

功能介绍

更新公网NAT网关实例。

接口约束

在admin_state_up = True & status = ACTIVE 时允许更新，支持更新名称、描述、规格。"

调用方法

请参见**如何调用API**。

URI

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

表 4-17 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-18 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-19 请求 Body 参数

参数	是否必选	参数类型	描述
nat_gateway	是	**UpdateNatGatewayOption** object	更新公网NAT网关实例的请求体。

表 4-20 UpdateNatGatewayOption

参数	是否必选	参数类型	描述
name	否	String	公网NAT网关实例的名字，长度限制为64。 公网NAT网关实例的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。 最小长度：**1** 最大长度：**64**
description	否	String	公网NAT网关的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	是否必选	参数类型	描述
spec	否	String	公网NAT网关的规格。 取值为： "1"：小型，SNAT最大连接数10000 "2"：中型，SNAT最大连接数50000 "3"：大型，SNAT最大连接数200000 "4"：超大型，SNAT最大连接数1000000 枚举值： • **1** • **2** • **3** • **4**
session_conf	否	**SessionConfiguration** object	公网NAT网关会话参数配置。

表 4-21 SessionConfiguration

参数	是否必选	参数类型	描述
tcp_session_expire_time	否	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	否	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	否	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**
tcp_time_wait_time	否	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

响应参数

状态码： 200

表 4-22 响应 Body 参数

参数	参数类型	描述
nat_gateway	**NatGatewayResponseBody** object	公网NAT网关实例的响应体。

表 4-23 NatGatewayResponseBody

参数	参数类型	描述
id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
name	String	公网NAT网关实例的名字，长度限制为64。 最小长度：**1** 最大长度：**64**
description	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	公网NAT网关的规格。 取值为： “1”：小型，SNAT最大连接数10000 “2”：中型，SNAT最大连接数50000 “3”：大型，SNAT最大连接数200000 “4”：超大型，SNAT最大连接数1000000 枚举值： • **1** • **2** • **3** • **4**

参数	参数类型	描述
status	String	公网NAT网关实例的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE": 创建中 "PENDING_UPDATE": 更新中 "PENDING_DELETE": 删除中 "INACTIVE": 不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： • "true": 解冻 • "false": 冻结
created_at	String	公网NAT网关实例的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
router_id	String	VPC的id。 最小长度：**36** 最大长度：**36**
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。 创建公网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
session_conf	**SessionConfiguration** object	公网NAT网关会话参数配置。
ngport_ip_address	String	公网NAT网关私有IP地址，由VPC中子网分配。 最小长度：**7** 最大长度：**15**
billing_info	String	订单信息。此字段只有在订购包周期资源时才会有订单信息，而在订购按需资源时则为空。

参数	参数类型	描述
dnat_rules_limit	Long	公网NAT网关下DNAT规则数量限制，默认为200。 最小值：**1** 最大值：**100000**
snat_rule_public_ip_limit	Integer	公网NAT网关下SNAT规则EIP池中EIP数量限制，默认为20。 最小值：**1** 最大值：**100**

表 4-24 SessionConfiguration

参数	参数类型	描述
tcp_session_expire_time	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**
tcp_time_wait_time	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

请求示例

```
PUT https://{Endpoint}/v2/70505c941b9b4dfd82fd351932328a2f/nat_gateways/
14338426-6afe-4019-996b-3a9525296e11

{
  "nat_gateway" : {
    "name" : "new_name",
    "description" : "new description",
    "spec" : "1"
  }
}
```

响应示例

状态码： 200

更新公网NAT网关实例成功。

```
{
  "nat_gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "new_name",
    "description" : "new description",
    "spec" : "1",
    "tenant_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22 08:47:13.238743",
    "internal_network_id" : "89d66639-aacb-4929-969d-07080b0f9fd9",
    "router_id" : "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
    "admin_state_up" : true,
    "billing_info" : "",
    "dnat_rules_limit" : 200,
    "snat_rule_public_ip_limit" : 20,
    "ngport_ip_address" : "192.168.0.138"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class UpdateNatGatewaySolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateNatGatewayRequest request = new UpdateNatGatewayRequest();
        UpdateNatGatewayRequestBody body = new UpdateNatGatewayRequestBody();
        UpdateNatGatewayOption natGatewaybody = new UpdateNatGatewayOption();
        natGatewaybody.withName("new_name")
            .withDescription("new description")
            .withSpec(UpdateNatGatewayOption.SpecEnum.fromValue("1"));
        body.withNatGateway(natGatewaybody);
        request.withBody(body);
        try {
            UpdateNatGatewayResponse response = client.updateNatGateway(request);
            System.out.println(response.toString());
```

```
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateNatGatewayRequest()
        natGatewaybody = UpdateNatGatewayOption(
            name="new_name",
            description="new description",
            spec="1"
        )
        request.body = UpdateNatGatewayRequestBody(
            nat_gateway=natGatewaybody
        )
        response = client.update_nat_gateway(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)
```

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNatGatewayRequest{}
    nameNatGateway:= "new_name"
    descriptionNatGateway:= "new description"
    specNatGateway:= model.GetUpdateNatGatewayOptionSpecEnum().E_1
    natGatewaybody := &model.UpdateNatGatewayOption{
        Name: &nameNatGateway,
        Description: &descriptionNatGateway,
        Spec: &specNatGateway,
    }
    request.Body = &model.UpdateNatGatewayRequestBody{
        NatGateway: natGatewaybody,
    }
    response, err := client.UpdateNatGateway(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新公网NAT网关实例成功。

错误码

请参见**错误码**。

4.1.5 查询指定的公网 NAT 网关详情

功能介绍

查询指定的公网NAT网关实例详情。

调用方法

请参见**如何调用API**。

URI

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

表 4-25 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-26 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-27 响应 Body 参数

参数	参数类型	描述
nat_gateway	**NatGatewayResponseBody** object	公网NAT网关实例的响应体。

表 4-28 NatGatewayResponseBody

参数	参数类型	描述
id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
name	String	公网NAT网关实例的名字，长度限制为64。 最小长度：**1** 最大长度：**64**
description	String	公网NAT网关实例的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	公网NAT网关的规格。 取值为： “1”：小型，SNAT最大连接数10000 “2”：中型，SNAT最大连接数50000 “3”：大型，SNAT最大连接数200000 “4”：超大型，SNAT最大连接数1000000 枚举值： ● **1** ● **2** ● **3** ● **4**
status	String	公网NAT网关实例的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "INACTIVE"：不可用 枚举值： ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： ● "true"：解冻 ● "false"：冻结

参数	参数类型	描述
created_at	String	公网NAT网关实例的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
router_id	String	VPC的id。 最小长度：**36** 最大长度：**36**
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。 创建公网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
session_conf	**SessionConfiguration** object	公网NAT网关会话参数配置。
ngport_ip_address	String	公网NAT网关私有IP地址，由VPC中子网分配。 最小长度：**7** 最大长度：**15**
billing_info	String	订单信息。此字段只有在订购包周期资源时才会有订单信息，而在订购按需资源时则为空。
dnat_rules_limit	Long	公网NAT网关下DNAT规则数量限制，默认为200。 最小值：**1** 最大值：**100000**
snat_rule_public_ip_limit	Integer	公网NAT网关下SNAT规则EIP池中EIP数量限制，默认为20。 最小值：**1** 最大值：**100**

表 4-29 SessionConfiguration

参数	参数类型	描述
tcp_session_expire_time	Integer	TCP会话过期时间。 最小值：**40** 最大值：**7200**
udp_session_expire_time	Integer	UDP会话过期时间。 最小值：**40** 最大值：**7200**
icmp_session_expire_time	Integer	ICMP会话过期时间。 最小值：**10** 最大值：**7200**
tcp_time_wait_time	Integer	TCP连接关闭时TIME_WAIT状态持续时间。 最小值：**0** 最大值：**1800**

请求示例

```
GET https://{Endpoint}/v2/70505c941b9b4dfd82fd351932328a2f/nat_gateways/
14338426-6afe-4019-996b-3a9525296e11
```

响应示例

状态码： 200

查询公网NAT网关实例成功。

```
{
  "nat_gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "nat-gateway-name",
    "description" : "nat-gateway-description",
    "spec" : "1",
    "router_id" : "d84f345c-80a1-4fa2-a39c-d0d397c3f09a",
    "tenant_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "internal_network_id" : "89d66639-aacb-4929-969d-07080b0f9fd9",
    "status" : "ACTIVE",
    "admin_state_up" : true,
    "billing_info" : "",
    "dnat_rules_limit" : 200,
    "snat_rule_public_ip_limit" : 20,
    "ngport_ip_address" : "192.168.0.138",
    "created_at" : "2019-04-22 08:47:13.902312"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ShowNatGatewaySolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowNatGatewayRequest request = new ShowNatGatewayRequest();
        try {
            ShowNatGatewayResponse response = client.showNatGateway(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

```
    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询公网NAT网关实例成功。

错误码

请参见**错误码**。

4.2 DNAT 规则

4.2.1 查询 DNAT 规则列表

功能介绍

查询DNAT规则列表。

接口约束

可以在URI后面用'?'和'&'添加不同的查询条件组合，支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/dnat_rules

表 4-30 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 4-31 Query 参数

参数	是否必选	参数类型	描述
admin_state_up	否	Boolean	解冻/冻结状态。 取值范围： "true"：解冻 "false"：冻结

参数	是否必选	参数类型	描述
external_service_port	否	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
floating_ip_address	否	String	弹性公网的IP地址。 最小长度：**7** 最大长度：**15**
status	否	Array	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 数组长度：**1 - 10** 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**
floating_ip_id	否	String	弹性公网IP的id。 最小长度：**36** 最大长度：**36**
internal_service_port	否	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
id	否	String	DNAT规则的ID。 最小长度：**1** 最大长度：**36**
description	否	String	DNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
created_at	否	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
nat_gateway_id	否	Array	公网NAT网关实例的ID。 数组长度：**1 - 10**
port_id	否	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	否	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
protocol	否	Array	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 数组长度：**1 - 10**

参数	是否必选	参数类型	描述
marker	否	String	分页查询的起始资源ID，表示从指定资源的下一条记录开始查询。 ● 若不传入marker和limit参数，查询结果返回第一页全部资源记录（默认2000条）。 ● 若不传入marker参数，limit为10，查询结果返回第1~10条资源记录。 ● 若marker为第10条记录的资源ID，limit为10，查询结果返回第11~20条资源记录。 ● 若marker为第10条记录的资源ID，不传入limit参数，查询结果返回第11条及之后的资源记录（默认2000条）。 最小值：**36** 最大值：**36**

请求参数

表 4-32 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-33 响应 Body 参数

参数	参数类型	描述
dnat_rules	Array of **NatGatewayDnatRuleResponseBody** objects	查询DNAT规则列表的响应体。 数组长度：**0 - 2000**

表 4-34 NatGatewayDnatRuleResponseBody

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
port_id	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**

参数	参数类型	描述
floating_ip_id	String	弹性公网IP的id。 最小长度：**1** 最大长度：**36**
floating_ip_address	String	弹性公网IP的IP地址。 最小长度：**7** 最大长度：**15**
external_service_port	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。
status	String	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **EIP_FREEZED** ● **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： - “true”： 解冻 - “false”： 冻结
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围
protocol	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： ● **tcp** ● **udp** ● **any**

参数	参数类型	描述
created_at	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
global_eip_id	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**
global_eip_address	String	全域弹性公网IP的地址。 最小长度：**7** 最大长度：**15**

请求示例

```
GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/dnat_rules?limit=2
```

响应示例

状态码： 200

查询DNAT规则列表成功。

```
{
  "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" : "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" : "d199ba7e0ba64899b2e81518104b1526d",
    "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"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListNatGatewayDnatRulesSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListNatGatewayDnatRulesRequest request = new ListNatGatewayDnatRulesRequest();
        request.withAdminStateUp(<admin_state_up>);
        request.withExternalServicePort(<external_service_port>);
        request.withFloatingIpAddress("<floating_ip_address>");
        request.withStatus();
        request.withFloatingIpId("<floating_ip_id>");
        request.withInternalServicePort(<internal_service_port>);
        request.withLimit(<limit>);
        request.withId("<id>");
        request.withDescription("<description>");
        request.withCreatedAt("<created_at>");
        request.withNatGatewayId();
        request.withPortId("<port_id>");
        request.withPrivateIp("<private_ip>");
        request.withProtocol();
        request.withMarker("<marker>");
        try {
            ListNatGatewayDnatRulesResponse response = client.listNatGatewayDnatRules(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```

```
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListNatGatewayDnatRulesRequest()
        request.admin_state_up = <AdminStateUp>
        request.external_service_port = <external_service_port>
        request.floating_ip_address = "<floating_ip_address>"
        request.status =
        request.floating_ip_id = "<floating_ip_id>"
        request.internal_service_port = <internal_service_port>
        request.limit = <limit>
        request.id = "<id>"
        request.description = "<description>"
        request.created_at = "<created_at>"
        request.nat_gateway_id =
        request.port_id = "<port_id>"
        request.private_ip = "<private_ip>"
        request.protocol =
        request.marker = "<marker>"
        response = client.list_nat_gateway_dnat_rules(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNatGatewayDnatRulesRequest{}
    adminStateUpRequest:= <admin_state_up>
    request.AdminStateUp = &adminStateUpRequest
    externalServicePortRequest:= int32(<external_service_port>)
    request.ExternalServicePort = &externalServicePortRequest
    floatingIpAddressRequest:= "<floating_ip_address>"
    request.FloatingIpAddress = &floatingIpAddressRequest
    floatingIpIdRequest:= "<floating_ip_id>"
    request.FloatingIpId = &floatingIpIdRequest
    internalServicePortRequest:= int32(<internal_service_port>)
    request.InternalServicePort = &internalServicePortRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    idRequest:= "<id>"
    request.Id = &idRequest
    descriptionRequest:= "<description>"
    request.Description = &descriptionRequest
    createdAtRequest:= "<created_at>"
    request.CreatedAt = &createdAtRequest
    portIdRequest:= "<port_id>"
    request.PortId = &portIdRequest
    privateIpRequest:= "<private_ip>"
    request.PrivateIp = &privateIpRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    response, err := client.ListNatGatewayDnatRules(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询DNAT规则列表成功。

错误码

请参见**错误码**。

4.2.2 创建 DNAT 规则

功能介绍

创建DNAT规则。

接口约束

创建规则时，要求网关状态status = ACTIVE，要求网关管理员状态admin_state_up = True。 port_id和private_ip不能同时生效。 对于all port类型的规则，要求internal_service_port = 0，external_service_port = 0，protocol = ANY.

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/dnat_rules

表 4-35 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 4-36 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-37 请求 Body 参数

参数	是否必选	参数类型	描述
dnat_rule	是	**CreateNatGatewayDnatOption** object	创建DNAT规则的请求体。

表 4-38 CreateNatGatewayDnatOption

参数	是否必选	参数类型	描述
description	否	String	DNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
port_id	否	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	否	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
internal_service_port	是	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535**
floating_ip_id	是	String	弹性公网IP的id。 最小长度：**36** 最大长度：**36**
external_service_port	是	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535**

参数	是否必选	参数类型	描述
protocol	是	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。
internal_service_port_range	否	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	否	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
global_eip_id	否	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**

响应参数

状态码： 201

表 4-39 响应 Body 参数

参数	参数类型	描述
dnat_rule	**NatGatewayDnatRuleResponseBody** object	DNAT规则的响应体。

表 4-40 NatGatewayDnatRuleResponseBody

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
port_id	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
floating_ip_id	String	弹性公网IP的id。 最小长度：**1** 最大长度：**36**
floating_ip_address	String	弹性公网IP的IP地址。 最小长度：**7** 最大长度：**15**
external_service_port	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。

参数	参数类型	描述
status	String	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： - “true”： 解冻 - “false”： 冻结
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围
protocol	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
created_at	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
global_eip_id	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
global_eip_address	String	全域弹性公网IP的地址。 最小长度：**7** 最大长度：**15**

请求示例

创建dnat规则。

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

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

响应示例

状态码： 201

创建DNAT规则成功。

```
{
  "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" : "d199ba7e0ba64899b2e81518104b1526d",
    "created_at" : "2019-11-15 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"
  }
}
```

状态码

状态码	描述
201	创建DNAT规则成功。

错误码

请参见**错误码**。

4.2.3 删除 DNAT 规则

功能介绍

删除指定的DNAT规则。

调用方法

请参见**如何调用API**。

URI

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

表 4-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-42 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/nat_gateways/f4dfea98-874a-46f7-aa2a-fb348d0ceb02/dnat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8"
```

响应示例

无

状态码

状态码	描述
204	删除DNAT规则成功。

错误码

请参见**错误码**。

4.2.4 更新 DNAT 规则

功能介绍

更新指定的DNAT规则。

接口约束

更新操作时，要求DNAT规则状态status = ACTIVE，要求网关管理员状态admin_state_up = True。port_id和private_ip不能同时生效。对于all port类型的规则，要求internal_service_port = 0，external_service_port = 0，protocol = ANY。 更新操作涉及以下字段更新时，要求这些字段必须一起更新。包括：port_id、private_ip、internal_service_port、external_service_port、floating_ip_id、protocol、internal_service_port_range、external_service_port_range。

调用方法

请参见**如何调用API**。

URI

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

表 4-43 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-44 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-45 请求 Body 参数

参数	是否必选	参数类型	描述
dnat_rule	是	**UpdateNatGatewayDnatRuleOption** object	更新DNAT规则的请求体。

表 4-46 UpdateNatGatewayDnatRuleOption

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	NAT网关的id。 最小长度：**36** 最大长度：**36**

参数	是否必选	参数类型	描述
description	否	String	DNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
port_id	否	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	否	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
protocol	否	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **TCP** • **UDP** • **ANY**
floating_ip_id	否	String	弹性公网IP的id。 最小长度：**36** 最大长度：**36**
internal_service_port	否	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**

参数	是否必选	参数类型	描述
external_service_port	否	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**10**
internal_service_port_range	否	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	否	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
global_eip_id	否	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**

响应参数

状态码： 200

表 4-47 响应 Body 参数

参数	参数类型	描述
dnat_rule	**NatGatewayDnatRuleResponseBody** object	DNAT规则的响应体。

表 4-48 NatGatewayDnatRuleResponseBody

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
port_id	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
floating_ip_id	String	弹性公网IP的id。 最小长度：**1** 最大长度：**36**
floating_ip_address	String	弹性公网IP的IP地址。 最小长度：**7** 最大长度：**15**
external_service_port	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。

参数	参数类型	描述
status	String	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： - “true”： 解冻 - “false”： 冻结
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围
protocol	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
created_at	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
global_eip_id	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
global_eip_address	String	全域弹性公网IP的地址。 最小长度：**7** 最大长度：**15**

请求示例

```
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-rules"
  }
}
```

响应示例

状态码： 200

更新DNAT规则成功。

```
{
  "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" : "d199ba7e0ba64899b2e81518104b1526",
    "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"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class UpdateNatGatewayDnatRuleSolution {
```



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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateNatGatewayDnatRuleRequest request = new UpdateNatGatewayDnatRuleRequest();
        UpdateNatGatewayDnatRuleRequestBody body = new UpdateNatGatewayDnatRuleRequestBody();
        UpdateNatGatewayDnatRuleOption dnatRulebody = new UpdateNatGatewayDnatRuleOption();
        dnatRulebody.withNatGatewayId("a78fb3eb-1654-4710-8742-3fc49d5f04f8")
            .withDescription("my dnat-rules");
        body.withDnatRule(dnatRulebody);
        request.withBody(body);
        try {
            UpdateNatGatewayDnatRuleResponse response = client.updateNatGatewayDnatRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateNatGatewayDnatRuleRequest()
```

```
        dnatRulebody = UpdateNatGatewayDnatRuleOption(
            nat_gateway_id="a78fb3eb-1654-4710-8742-3fc49d5f04f8",
            description="my dnat-rules"
        )
        request.body = UpdateNatGatewayDnatRuleRequestBody(
            dnat_rule=dnatRulebody
        )
        response = client.update_nat_gateway_dnat_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNatGatewayDnatRuleRequest{}
    descriptionDnatRule:= "my dnat-rules"
    dnatRulebody := &model.UpdateNatGatewayDnatRuleOption{
        NatGatewayId: "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        Description: &descriptionDnatRule,
    }
    request.Body = &model.UpdateNatGatewayDnatRuleRequestBody{
        DnatRule: dnatRulebody,
    }
    response, err := client.UpdateNatGatewayDnatRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新DNAT规则成功。

错误码

请参见**错误码**。

4.2.5 查询指定的 DNAT 规则详情

功能介绍

查询指定的DNAT规则详情。

调用方法

请参见**如何调用API**。

URI

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

表 4-49 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-50 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-51 响应 Body 参数

参数	参数类型	描述
dnat_rule	**NatGatewayDnatRuleResponseBody** object	DNAT规则的响应体。

表 4-52 NatGatewayDnatRuleResponseBody

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	参数类型	描述
port_id	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
floating_ip_id	String	弹性公网IP的id。 最小长度：**1** 最大长度：**36**
floating_ip_address	String	弹性公网IP的IP地址。 最小长度：**7** 最大长度：**15**
external_service_port	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。
status	String	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**

参数	参数类型	描述
admin_state_up	Boolean	解冻/冻结状态。 取值范围： - “true”： 解冻 - “false”： 冻结
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围
protocol	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
created_at	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
global_eip_id	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**
global_eip_address	String	全域弹性公网IP的地址。 最小长度：**7** 最大长度：**15**

请求示例

```
GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526d/dnat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
```

响应示例

状态码： 200

查询DNAT规则成功。

```
{
  "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",
  "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"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ShowNatGatewayDnatRuleSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowNatGatewayDnatRuleRequest request = new ShowNatGatewayDnatRuleRequest();
        try {
            ShowNatGatewayDnatRuleResponse response = client.showNatGatewayDnatRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
```

```
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
        Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询DNAT规则成功。

错误码

请参见**错误码**。

4.2.6 批量创建 DNAT 规则

功能介绍

批量创建DNAT规则。

接口约束

批量创建规则时，要求网关状态status = ACTIVE，要求网关管理员状态admin_state_up = True。 port_id和private_ip不能同时生效。 对于all port类型的规则，要求internal_service_port = 0，external_service_port = 0，protocol = ANY。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/dnat_rules/batch

表 4-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 4-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	否	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-55 请求 Body 参数

参数	是否必选	参数类型	描述
dnat_rules	是	Array of **CreateNatGatewayDnatOption** objects	DNAT规则批量创建对象的请求体。

表 4-56 CreateNatGatewayDnatOption

参数	是否必选	参数类型	描述
description	否	String	DNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	是否必选	参数类型	描述
port_id	否	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	否	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
internal_service_port	是	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535**
floating_ip_id	是	String	弹性公网IP的id。 最小长度：**36** 最大长度：**36**
external_service_port	是	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535**
protocol	是	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。
internal_service_port_range	否	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。

参数	是否必选	参数类型	描述
external_service_port_range	否	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal_service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
global_eip_id	否	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**

响应参数

状态码： 201

表 4-57 响应 Body 参数

参数	参数类型	描述
dnat_rules	Array of **NatGatewayDnatRuleResponseBody** objects	DNAT规则批量创建对象的响应体。

表 4-58 NatGatewayDnatRuleResponseBody

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	参数类型	描述
port_id	String	虚拟机或者裸机的Port ID，对应虚拟私有云场景，与private_ip参数二选一。 最小长度：**36** 最大长度：**36**
private_ip	String	用户私有IP地址，对应专线、云连接场景，与port_id参数二选一。 最小长度：**7** 最大长度：**15**
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
floating_ip_id	String	弹性公网IP的id。 最小长度：**1** 最大长度：**36**
floating_ip_address	String	弹性公网IP的IP地址。 最小长度：**7** 最大长度：**15**
external_service_port	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。
status	String	DNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**

参数	参数类型	描述
admin_state_up	Boolean	解冻/冻结状态。 取值范围： - “true”： 解冻 - “false”： 冻结
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 功能说明：该端口范围与external _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 功能说明：该端口范围与internal _service_port_range按顺序实现1:1映射。 取值范围：1~65535。 约束：只能以’-’字符连接端口范围
protocol	String	协议类型，目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
created_at	String	DNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
global_eip_id	String	全域弹性公网IP的id。 最小长度：**36** 最大长度：**36**
global_eip_address	String	全域弹性公网IP的地址。 最小长度：**7** 最大长度：**15**

请求示例

批量创建规则(第一条为指定端口的规则，第二条为all port类型的规则)。

```
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,
```



```
    "description" : "my dnat rule 01"
  }, {
    "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"
  } ]
}
```

响应示例

状态码： 201

POST操作正常返回。

```
{
  "dnat_rules" : [ {
    "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",
    "private_ip" : "",
    "internal_service_port" : 993,
    "protocol" : "tcp",
    "tenant_id" : "d199ba7e0ba64899b2e81518104b1526",
    "created_at" : "2019-11-15 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" : "PENDING_CREATE",
    "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" : "d199ba7e0ba64899b2e81518104b1526",
    "created_at" : "2019-11-15 15:44:42.595173",
    "id" : "79195d50-0271-41f1-bded-4c089c2502ff",
    "floating_ip_address" : "5.21.11.227",
    "external_service_port" : 0,
    "description" : "my dnat rule 01"
  } ]
}
```

状态码

状态码	描述
201	POST操作正常返回。

错误码

请参见**错误码**。

4.3 SNAT 规则

4.3.1 查询 SNAT 规则列表

功能介绍

查询SNAT规则列表。

接口约束

可以在URI后面用'?'和'&'添加不同的查询条件组合。支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/snat_rules

表 4-59 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 4-60 Query 参数

参数	是否必选	参数类型	描述
admin_state_up	否	Boolean	解冻/冻结状态。 取值范围： "true"：解冻 "false"：冻结
cidr	否	String	可以是网段或者主机格式，与network_id参数二选一。Source_type=0时，cidr必须是vpc子网网段的子集(不能相等）；Source_type=1时，cidr必须指定专线侧网段。

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
floating_ip_address	否	Array	功能说明：弹性公网IP。 数组长度：**1 - 20**
floating_ip_id	否	Array	功能说明：弹性公网IP的id。 数组长度：**1 - 20**
id	否	String	SNAT规则的ID。 最小长度：**1** 最大长度：**36**
description	否	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
created_at	否	String	SNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
nat_gateway_id	否	Array	公网NAT网关实例的ID。 数组长度：**1 - 10**
network_id	否	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
source_type	否	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**

参数	是否必选	参数类型	描述
status	否	String	SNAT规则的状态。 取值为: "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值: ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **EIP_FREEZED** ● **INACTIVE**
marker	否	String	分页查询的起始资源ID，表示从指定资源的下一条记录开始查询。 ● 若不传入marker和limit参数，查询结果返回第一页全部资源记录（默认2000条）。 ● 若不传入marker参数，limit为10，查询结果返回第1~10条资源记录。 ● 若marker为第10条记录的资源ID，limit为10，查询结果返回第11~20条资源记录。 ● 若marker为第10条记录的资源ID，不传入limit参数，查询结果返回第11条及之后的资源记录（默认2000条）。 最小值：**36** 最大值：**36**

请求参数

表 4-61 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-62 响应 Body 参数

参数	参数类型	描述
snat_rules	Array of **NatGatewaySnatRuleResponseBody** objects	查询SNAT规则列表的响应体。 数组长度：**2000 - 0**

表 4-63 NatGatewaySnatRuleResponseBody

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
cidr	String	cidr，可以是网段或者主机格式，与network_id参数二选一。 Source_type=0时，cidr必须是vpc子网网段的子集(不能相等）；Source_type=1时，cidr必须指定专线侧网段。 最小长度：**9** 最大长度：**18**
source_type	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**
floating_ip_id	String	功能说明：弹性公网IP的id，多个弹性公网IP使用逗号分隔。
description	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
status	String	SNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **EIP_FREEZED** ● **INACTIVE**
created_at	String	SNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
network_id	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： ● "true"：解冻 ● "false"：冻结

参数	参数类型	描述
floating_ip_address	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。
freezed_ip_address	String	功能说明：冻结的弹性公网IP，多个冻结的弹性公网IP使用逗号分隔。
global_eip_id	String	全域弹性公网IP的id。
global_eip_address	String	全域弹性公网IP的地址。

请求示例

```
GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules?limit=2
```

响应示例

状态码： 200

查询SNAT规则列表成功。

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

SDK 代码示例

SDK代码示例如下。

Java

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

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListNatGatewaySnatRulesSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListNatGatewaySnatRulesRequest request = new ListNatGatewaySnatRulesRequest();
        request.withAdminStateUp(<admin_state_up>);
        request.withCidr("<cidr>");
        request.withLimit(<limit>);
        request.withFloatingIpAddress();
        request.withFloatingIpId();
        request.withId("<id>");
        request.withDescription("<description>");
        request.withCreatedAt("<created_at>");
        request.withNatGatewayId();
        request.withNetworkId("<network_id>");
        request.withSourceType(<source_type>);
        request.withStatus(ListNatGatewaySnatRulesRequest.StatusEnum.fromValue("<status>"));
        request.withMarker("<marker>");
        try {
            ListNatGatewaySnatRulesResponse response = client.listNatGatewaySnatRules(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListNatGatewaySnatRulesRequest()
        request.admin_state_up = <AdminStateUp>
        request.cidr = "<cidr>"
        request.limit = <limit>
        request.floating_ip_address =
        request.floating_ip_id =
        request.id = "<id>"
        request.description = "<description>"
        request.created_at = "<created_at>"
        request.nat_gateway_id =
        request.network_id = "<network_id>"
        request.source_type = <source_type>
        request.status = "<status>"
        request.marker = "<marker>"
        response = client.list_nat_gateway_snat_rules(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

  request := &model.ListNatGatewaySnatRulesRequest{}
  adminStateUpRequest:= <admin_state_up>
  request.AdminStateUp = &adminStateUpRequest
  cidrRequest:= "<cidr>"
  request.Cidr = &cidrRequest
  limitRequest:= int32(<limit>)
  request.Limit = &limitRequest
  idRequest:= "<id>"
  request.Id = &idRequest
  descriptionRequest:= "<description>"
  request.Description = &descriptionRequest
  createdAtRequest:= "<created_at>"
  request.CreatedAt = &createdAtRequest
  networkIdRequest:= "<network_id>"
  request.NetworkId = &networkIdRequest
  sourceTypeRequest:= int32(<source_type>)
  request.SourceType = &sourceTypeRequest
  statusRequest:= model.GetListNatGatewaySnatRulesRequestStatusEnum().<STATUS>
  request.Status = &statusRequest
  markerRequest:= "<marker>"
  request.Marker = &markerRequest
  response, err := client.ListNatGatewaySnatRules(request)
  if err == nil {
     fmt.Printf("%+v\n", response)
  } else {
     fmt.Println(err)
  }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询SNAT规则列表成功。

错误码

请参见**错误码**。

4.3.2 创建 SNAT 规则

功能介绍

创建SNAT规则。

接口约束

创建规则时，要求网关状态status = ACTIVE，要求网关管理员状态admin_state_up = True

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/snat_rules

表 4-64 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 4-65 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-66 请求 Body 参数

参数	是否必选	参数类型	描述
snat_rule	是	**CreateNatGatewaySnatRuleOption** object	创建SNAT规则的请求体。

表 4-67 CreateNatGatewaySnatRuleOption

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	否	String	cidr，可以是网段或者主机格式，与network_id参数二选一。Source_type=0时，cidr必须是vpc 子网网段的子集(不能相等）; Source_type=1时，cidr必须指定专线侧网段。 最小长度：**9** 最大长度：**18**
network_id	否	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	否	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
source_type	否	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**
floating_ip_id	是	String	功能说明：弹性公网IP的id，多个弹性公网IP使用逗号分隔。 约束：弹性公网IP的id个数不能超过20个。
global_eip_id	否	String	全域弹性公网IP的id。

响应参数

状态码： 201

表 4-68 响应 Body 参数

参数	参数类型	描述
snat_rule	**CreateNatGatewaySnatRuleResponseBody** object	创建SNAT规则的响应体。

表 4-69 CreateNatGatewaySnatRuleResponseBody

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	cidr，可以是网段或者主机格式，与network_id参数二选一。 Source_type=0时，cidr必须是vpc子网网段的子集(不能相等）；Source_type=1时，cidr必须指定专线侧网段。 最小长度：**9** 最大长度：**18**
source_type	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**
floating_ip_id	String	功能说明：弹性公网IP的id，多个弹性公网IP使用逗号分隔。
description	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	参数类型	描述
status	String	SNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**
created_at	String	SNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
network_id	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： • "true"：解冻 • "false"：冻结
floating_ip_address	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。
global_eip_id	String	全域弹性公网IP的id。
global_eip_address	String	全域弹性公网IP的地址。

请求示例

SNAT规则创建的请求体。

```
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"
  }
}
```

响应示例

状态码： 201

创建SNAT规则成功。

```
{
  "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"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

SNAT规则创建的请求体。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreateNatGatewaySnatRuleSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateNatGatewaySnatRuleRequest request = new CreateNatGatewaySnatRuleRequest();
        CreateNatGatewaySnatRuleRequestOption body = new CreateNatGatewaySnatRuleRequestOption();
        CreateNatGatewaySnatRuleOption snatRulebody = new CreateNatGatewaySnatRuleOption();
        snatRulebody.withNatGatewayId("a78fb3eb-1654-4710-8742-3fc49d5f04f8")
            .withCidr("172.30.0.0/24")
            .withDescription("my snat rule 01")
```

```
            .withSourceType(1)
            .withFloatingIpId("bdc10a4c-d81a-41ec-adf7-de857f7c812a");
        body.withSnatRule(snatRulebody);
        request.withBody(body);
        try {
            CreateNatGatewaySnatRuleResponse response = client.createNatGatewaySnatRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

SNAT规则创建的请求体。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateNatGatewaySnatRuleRequest()
        snatRulebody = CreateNatGatewaySnatRuleOption(
            nat_gateway_id="a78fb3eb-1654-4710-8742-3fc49d5f04f8",
            cidr="172.30.0.0/24",
            description="my snat rule 01",
            source_type=1,
            floating_ip_id="bdc10a4c-d81a-41ec-adf7-de857f7c812a"
        )
        request.body = CreateNatGatewaySnatRuleRequestOption(
            snat_rule=snatRulebody
        )
        response = client.create_nat_gateway_snat_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

SNAT规则创建的请求体。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNatGatewaySnatRuleRequest{}
    cidrSnatRule:= "172.30.0.0/24"
    descriptionSnatRule:= "my snat rule 01"
    sourceTypeSnatRule:= int32(1)
    snatRulebody := &model.CreateNatGatewaySnatRuleOption{
        NatGatewayId: "a78fb3eb-1654-4710-8742-3fc49d5f04f8",
        Cidr: &cidrSnatRule,
        Description: &descriptionSnatRule,
        SourceType: &sourceTypeSnatRule,
        FloatingIpId: "bdc10a4c-d81a-41ec-adf7-de857f7c812a",
    }
    request.Body = &model.CreateNatGatewaySnatRuleRequestOption{
        SnatRule: snatRulebody,
    }
    response, err := client.CreateNatGatewaySnatRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建SNAT规则成功。

错误码

请参见**错误码**。

4.3.3 更新 SNAT 规则

功能介绍

更新指定的SNAT规则。

接口约束

更新弹性公网IP的id时，要求SNAT规则状态status = ACTIVE，要求网关管理员状态admin_state_up = True。 更新描述时，要求SNAT规则状态status = ACTIVE，要求网关管理员状态admin_state_up = True。

调用方法

请参见**如何调用API**。

URI

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

表 4-70 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-71 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 4-72 请求 Body 参数

参数	是否必选	参数类型	描述
snat_rule	是	**UpdateNatGatewaySnatRuleOption** object	更新SNAT规则的请求体。

表 4-73 UpdateNatGatewaySnatRuleOption

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	公网NAT网关的id。 最小长度：**36** 最大长度：**36**
public_ip_address	否	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。 约束：弹性公网IP的id个数不能超过20个
global_eip_id	否	String	全域弹性公网IP的id。
description	否	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

响应参数

状态码： 200

表 4-74 响应 Body 参数

参数	参数类型	描述
snat_rule	**NatGatewayUpdateSnatRuleResponseBody** object	更新SNAT规则的响应体。

表 4-75 NatGatewayUpdateSnatRuleResponseBody

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
source_type	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**
cidr	String	cidr，可以是网段或者主机格式，与network_id参数二选一。 Source_type=0时，cidr必须是vpc子网网段的子集(不能相等）；Source_type=1时，cidr必须指定专线侧网段。 最小长度：**9** 最大长度：**18**
floating_ip_id	String	功能说明：弹性公网IP的id，多个弹性公网IP使用逗号分隔。
description	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	参数类型	描述
status	String	SNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： • **ACTIVE** • **PENDING_CREATE** • **PENDING_UPDATE** • **PENDING_DELETE** • **EIP_FREEZED** • **INACTIVE**
created_at	String	SNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
network_id	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： • "true"：解冻 • "false"：冻结
floating_ip_address	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。
public_ip_address	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。
global_eip_address	String	全域弹性公网IP的地址。
global_eip_id	String	全域弹性公网IP的id。

请求示例

SNAT规则更新的请求体。

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

{
  "snat_rule" : {
    "nat_gateway_id" : "bbe7c2e7-3bad-445b-a067-b30acce66053",
    "description" : "my_snat_rule_update",
    "public_ip_address" : "10.15.10.11,10.15.10.12"
```

```
  }
}
```

响应示例

状态码： 200

PUT操作正常返回。

```
{
  "snat_rule" : {
    "floating_ip_id" : " bdc10a4c-d81a-41ec-adf7-de857f7c812a,7a094014-9657-463f-972b-e84d56b931a0",
    "status" : "ACTIVE",
    "nat_gateway_id" : "bbe7c2e7-3bad-445b-a067-b30acce66053",
    "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_update"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

SNAT规则更新的请求体。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class UpdateNatGatewaySnatRuleSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateNatGatewaySnatRuleRequest request = new UpdateNatGatewaySnatRuleRequest();
```

```
        UpdateNatGatewaySnatRuleRequestOption body = new UpdateNatGatewaySnatRuleRequestOption();
        UpdateNatGatewaySnatRuleOption snatRulebody = new UpdateNatGatewaySnatRuleOption();
        snatRulebody.withNatGatewayId("bbe7c2e7-3bad-445b-a067-b30acce66053")
            .withPublicIpAddress("10.15.10.11,10.15.10.12")
            .withDescription("my_snat_rule_update");
        body.withSnatRule(snatRulebody);
        request.withBody(body);
        try {
            UpdateNatGatewaySnatRuleResponse response = client.updateNatGatewaySnatRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

SNAT规则更新的请求体。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateNatGatewaySnatRuleRequest()
        snatRulebody = UpdateNatGatewaySnatRuleOption(
            nat_gateway_id="bbe7c2e7-3bad-445b-a067-b30acce66053",
            public_ip_address="10.15.10.11,10.15.10.12",
            description="my_snat_rule_update"
        )
        request.body = UpdateNatGatewaySnatRuleRequestOption(
            snat_rule=snatRulebody
        )
        response = client.update_nat_gateway_snat_rule(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

SNAT规则更新的请求体。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateNatGatewaySnatRuleRequest{}
    publicIpAddressSnatRule:= "10.15.10.11,10.15.10.12"
    descriptionSnatRule:= "my_snat_rule_update"
    snatRulebody := &model.UpdateNatGatewaySnatRuleOption{
        NatGatewayId: "bbe7c2e7-3bad-445b-a067-b30acce66053",
        PublicIpAddress: &publicIpAddressSnatRule,
        Description: &descriptionSnatRule,
    }
    request.Body = &model.UpdateNatGatewaySnatRuleRequestOption{
        SnatRule: snatRulebody,
    }
    response, err := client.UpdateNatGatewaySnatRule(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	PUT操作正常返回。

错误码

请参见**错误码**。

4.3.4 查询指定的 SNAT 规则详情

功能介绍

查询指定的SNAT规则详情。

接口约束

null

调用方法

请参见**如何调用API**。

URI

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

表 4-76 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-77 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 4-78 响应 Body 参数

参数	参数类型	描述
snat_rule	**NatGatewaySnatRuleResponseBody** object	SNAT规则的响应体。

表 4-79 NatGatewaySnatRuleResponseBody

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
tenant_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
nat_gateway_id	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	cidr，可以是网段或者主机格式，与network_id参数二选一。 Source_type=0时，cidr必须是vpc子网网段的子集(不能相等）；Source_type=1时，cidr必须指定专线侧网段。 最小长度：**9** 最大长度：**18**
source_type	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC） 最小值：**0** 最大值：**1** 缺省值：**0**
floating_ip_id	String	功能说明：弹性公网IP的id，多个弹性公网IP使用逗号分隔。

参数	参数类型	描述
description	String	SNAT规则的描述，长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
status	String	SNAT规则的状态。 取值为： "ACTIVE": 可用 "PENDING_CREATE"：创建中 "PENDING_UPDATE"：更新中 "PENDING_DELETE"：删除中 "EIP_FREEZED"：EIP冻结 "INACTIVE"：不可用 枚举值： ● **ACTIVE** ● **PENDING_CREATE** ● **PENDING_UPDATE** ● **PENDING_DELETE** ● **EIP_FREEZED** ● **INACTIVE**
created_at	String	SNAT规则的创建时间，格式是yyyy-mm-dd hh:mm:ss.SSSSSS。 最小长度：**1** 最大长度：**36**
network_id	String	规则使用的网络id。与cidr参数二选一。 最小长度：**36** 最大长度：**36**
admin_state_up	Boolean	解冻/冻结状态。 取值范围： ● "true"：解冻 ● "false"：冻结
floating_ip_address	String	功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。
freezed_ip_address	String	功能说明：冻结的弹性公网IP，多个冻结的弹性公网IP使用逗号分隔。
global_eip_id	String	全域弹性公网IP的id。
global_eip_address	String	全域弹性公网IP的地址。

请求示例

```
GET https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d33
```

响应示例

状态码： 200

查询SNAT规则成功。

```
{
  "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"
  }
}
```

状态码

状态码	描述
200	查询SNAT规则成功。

错误码

请参见**错误码**。

4.3.5 删除 SNAT 规则

功能介绍

删除指定的SNAT规则。

调用方法

请参见**如何调用API**。

URI

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

表 4-80 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	公网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 4-81 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v2/d199ba7e0ba64899b2e81518104b1526/nat_gateways/f4dfea98-874a-46f7-aa2a-fb348d0ceb02/snat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeleteNatGatewaySnatRuleSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteNatGatewaySnatRuleRequest request = new DeleteNatGatewaySnatRuleRequest();
        try {
            DeleteNatGatewaySnatRuleResponse response = client.deleteNatGatewaySnatRule(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

    client = NatClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(NatRegion.value_of("<YOUR REGION>")) \
        .build()
```

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除SNAT规则成功。

错误码

请参见**错误码**。

4.4 公网 NAT 网关标签 v2.0

4.4.1 查询公网 NAT 网关资源实例

功能介绍

- 使用标签过滤公网NAT网关资源实例。
- 标签管理服务需要提供按标签过滤公网NAT网关服务实例并汇总显示在列表中，需要公网NAT网关服务提供查询能力。
- 资源默认按照创建时间倒序，资源tag也按照创建时间倒序。

接口约束

- 需使用VPC域名调用公网NAT标签API。

调用方法

请参见**如何调用API**。

URI

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

表 4-82 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。

请求参数

表 4-83 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-84 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of **Tag** objects	包含标签对象列表，最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：**1 - 10**
tags_any	否	Array of **Tag** objects	包含任意标签对象列表，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。
not_tags	否	Array of **Tag** objects	不包含标签对象列表，最多包含10个key，每个key下面的value最多10个, 结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。
not_tags_any	否	Array of **Tag** objects	不包含任意标签对象列表，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。
limit	否	String	查询记录数（action为count时无此参数）如果action为filter默认为1000，limit最多为1000,不能为负数，最小值为1

参数	是否必选	参数类型	描述
offset	否	String	（索引位置）， 从offset指定的下一条数据开始查询。查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据时响应体中的值带入此参数（action为count时无此参数）如果action为filter默认为0，必须为数字，不能为负数
action	是	String	• 操作标识（仅限于filter，count）： filter（过滤），count（查询总条数） • 如果是filter就是分页查询，如果是count只需按照条件将总条数返回即可。
matches	否	Array of **Match** objects	• 搜索字段列表，key为要匹配的字段，如resource_name等。value为匹配的值。此字段为固定字典值。 • 根据不同的字段确认是否需要模糊匹配，如resource_name默认为模糊搜索（不区分大小写），如果value为空字符串精确匹配。resource_id为精确匹配。

表 4-85 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。(搜索时不对此参数做校验)，key不能为空或者空字符串，不能为空格，校验和使用之前先trim 前后空格。

参数	是否必选	参数类型	描述
values	是	Array of strings	• 值列表。每个值最大长度255个unicode字符，不能为空格，校验和使用之前先trim前后空格。 • *为系统保留字符，value可为空但不可缺省。 • 如果里面的value是以*开头表示按照*后面的值全模糊匹配。 • 如果values为空列表，则表示any_value（查询任意value）。value之间为或的关系。

表 4-86 Match

参数	是否必选	参数类型	描述
key	是	String	键。限定为resource_name。
value	是	String	值。每个值最大长度255个unicode字符。

响应参数

状态码： 200

表 4-87 响应 Body 参数

参数	参数类型	描述
resources	Array of **Resource** objects	资源对象列表。请参考表Resource字段数据结构说明。
total_count	Integer	总记录数

表 4-88 Resource

参数	参数类型	描述
resource_id	String	资源ID
resource_detail	Object	资源详情。资源对象，用于扩展。默认为空

参数	参数类型	描述
tags	Array of ResourceTag objects	标签列表，没有标签默认为空数组。请参考表ResourceTag字段数据结构说明。
resource_name	String	资源名称，没有默认为空字符串

表 4-89 ResourceTag

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，*,<,>,,=
value	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，*,<,>,,=

请求示例

- 查询公网NAT网关资源实例，其中，操作标识为filter，进行分页查询，查询记录数为100条。

```
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" ]
  } ]
}
```

- 查询公网NAT网关资源实例，其中，操作标识为count，按照条件将总条数返回。

```
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" ]
 } ]
}
```

响应示例

状态码： 200

- 查询操作成功。
- 示例1：action为filter时的响应体
- 示例2：action为count时的响应体
- 示例 1

```
{
 "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
}
```

- 示例 2

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

状态码

状态码	描述
200	• 查询操作成功。 • 示例1：action为filter时的响应体 • 示例2：action为count时的响应体

错误码

请参见**错误码**。

4.4.2 批量添加/删除公网 NAT 网关资源标签

功能介绍

- 为指定公网NAT网关实例批量添加或删除标签。
- 标签管理服务需要使用该接口批量管理实例的标签。
- 一个资源上最多有10个标签。

接口约束

- 需使用VPC域名调用公网NAT标签API。
- 此接口为幂等接口：
  - 创建时如果请求体中存在重复key则报错。
  - 创建时，不允许重复key，如果数据库存在就覆盖。
  - 删除时，如果删除的标签不存在，默认处理成功，删除时不对标签字符集范围做校验。key最大长度为128个unicode字符，value最大长度为255个unicode字符。删除时tags结构体不能缺失，key不能为空，或者空字符串。

调用方法

请参见**如何调用API**。

URI

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

表 4-90 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	公网NAT网关ID。

请求参数

表 4-91 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-92 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of **Tags** objects	标签列表。请参考表Tags字段数据结构说明
action	是	String	操作标识：仅限于create（创建）、delete（删除）

表 4-93 Tags

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	是	String	值。每个值最大长度255个unicode字符，删除时如果value有值按照key/value删除，如果value没值，则按照key删除，可以为空字符串。,不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

响应参数

无

请求示例

- 批量添加公网NAT网关资源标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
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"
  } ]
}
```

- 批量删除公网NAT网关资源标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
{
  "action" : "delete",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

响应示例

无

状态码

状态码	描述
204	批量添加或删除操作成功。

错误码

请参见**错误码**。

4.4.3 添加公网 NAT 网关资源标签

功能介绍

- 添加公网NAT网关资源标签。一个资源上最多有10个标签。

接口约束

- 需使用VPC域名调用公网NAT标签API。

- 此接口为幂等接口：
  - 创建时，如果创建的标签已经存在（key相同），则覆盖。
- 创建tag时，要求网关存在。

调用方法

请参见**如何调用API**。

URI

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

表 4-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	所属公网NAT网关的id。

请求参数

表 4-95 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-96 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	**TagBody** object	标签列表。请参考表TagBody字段数据结构说明。

表 4-97 TagBody

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	是	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

响应参数

无

请求示例

添加公网NAT网关资源标签，其中，标签键为“key1”，对应的值为“value1”。

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

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

响应示例

无

状态码

状态码	描述
204	添加操作成功。

错误码

请参见**错误码**。

4.4.4 删除公网 NAT 网关资源标签

功能介绍

- 删除指定公网NAT网关资源实例的标签信息。

接口约束

- 需使用VPC域名调用公网NAT标签API。
- 此接口为幂等接口：
  - 删除时，不对标签做校验。删除的key不存在报404，key不能为空或者空字符串。

调用方法

请参见**如何调用API**。

URI

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

表 4-98 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	公网NAT网关id。
key	是	String	标签key。

请求参数

表 4-99 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

无

请求示例

```
DELETE  https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/
fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags/key1
```

响应示例

无

状态码

状态码	描述
204	删除操作成功。

错误码

请参见**错误码**。

4.4.5 查询公网 NAT 网关资源标签

功能介绍

- 查询指定公网NAT网关实例的标签信息。
- 标签管理服务需要使用该接口查询指定公网NAT网关实例的全部标签数据。

接口约束

- 需使用VPC域名调用公网NAT标签API。

调用方法

请参见**如何调用API**。

URI

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

表 4-100 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	公网NAT网关ID。

请求参数

表 4-101 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

状态码： 200

表 4-102 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagBody** objects	标签列表。

表 4-103 TagBody

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

请求示例

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

响应示例

状态码： 200

查询操作成功。

```
{
  "tags" : [ {
```

```
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

4.4.6 查询公网 NAT 网关项目标签

功能介绍

- 查询租户在指定项目和公网NAT网关实例类型的所有标签集合。
- 标签管理服务需要能够列出当前租户全部已使用的标签集合，为各服务Console打标签和过滤实例时提供标签联想功能。

接口约束

- 需使用VPC域名调用公网NAT标签API。

调用方法

请参见**如何调用API**。

URI

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

表 4-104 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。

请求参数

表 4-105 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

状态码： 200

表 4-106 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagsBody** objects	标签列表。

表 4-107 TagsBody

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，*,<,>,,=
values	Array of strings	值列表。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，*,<,>,,=

请求示例

```
GET https://{VPC_endpoint}/v2.0/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/tags
```

响应示例

状态码： 200

查询操作成功。

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

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

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

4.5 公网 NAT 网关标签 v3

4.5.1 查询公网 NAT 网关资源实例

功能介绍

- 使用标签过滤公网NAT网关资源实例。
- 标签管理服务需要提供按标签过滤公网NAT网关服务实例并汇总显示在列表中，需要公网NAT网关服务提供查询能力。
- 资源默认按照创建时间倒序，资源tag也按照创建时间倒序。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/nat_gateways/resource_instances/action

表 4-108 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。

请求参数

表 4-109 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-110 请求 Body 参数

参数	是否必选	参数类型	描述
tags	否	Array of **PublicTag** objects	包含标签对象列表，最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：**1 - 10**
tags_any	否	Array of **PublicTag** objects	包含任意标签对象列表，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。
not_tags	否	Array of **PublicTag** objects	不包含标签对象列表，最多包含10个key，每个key下面的value最多10个, 结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。

参数	是否必选	参数类型	描述
not_tags_any	否	Array of **PublicTag** objects	不包含任意标签对象列表，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。
limit	否	String	查询记录数（action为count时无此参数）如果action为filter默认为1000，limit最多为1000,不能为负数，最小值为1
offset	否	String	（索引位置）， 从offset指定的下一条数据开始查询。查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据时响应体中的值带入此参数（action为count时无此参数）如果action为filter默认为0，必须为数字，不能为负数
action	是	String	• 操作标识（仅限于filter，count）： filter（过滤），count（查询总条数） • 如果是filter就是分页查询，如果是count只需按照条件将总条数返回即可。
matches	否	Array of **PublicMatch** objects	• 搜索字段列表，key为要匹配的字段，如resource_name等。value为匹配的值。此字段为固定字典值。 • 根据不同的字段确认是否需要模糊匹配，如resource_name默认为模糊搜索（不区分大小写），如果value为空字符串精确匹配。resource_id为精确匹配。

表 4-111 PublicTag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。(搜索时不对此参数做校验)，key不能为空或者空字符串，不能为空格，校验和使用之前先trim 前后空格。
values	是	Array of strings	• 值列表。每个值最大长度255个unicode字符，不能为空格，校验和使用之前先trim前后空格。 • *为系统保留字符，value可为空但不可缺省。 • 如果里面的value是以*开头表示按照*后面的值全模糊匹配。 • 如果values为空列表，则表示any_value（查询任意value）。value之间为或的关系。

表 4-112 PublicMatch

参数	是否必选	参数类型	描述
key	是	String	键。限定为resource_name。
value	是	String	值。每个值最大长度255个unicode字符。

响应参数

状态码： 200

表 4-113 响应 Body 参数

参数	参数类型	描述
resources	Array of **PublicResource** objects	资源对象列表。请参考表Resource字段数据结构说明。
total_count	Integer	总记录数

表 4-114 PublicResource

参数	参数类型	描述
resource_id	String	资源ID
resource_detail	Object	资源详情。资源对象，用于扩展。默认为空
tags	Array of **PublicResourceTag** objects	标签列表，没有标签默认为空数组。请参考表ResourceTag字段数据结构说明。
resource_name	String	资源名称，没有默认为空字符串

表 4-115 PublicResourceTag

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，*,<,>,\,=
value	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，*,<,>,\,=

请求示例

- 查询公网NAT网关资源实例，其中，操作标识为filter，进行分页查询，查询记录数为100条。

```
POST https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/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" ]
```

```
 } ]
}
```

- 查询公网NAT网关资源实例，其中，操作标识为count，按照条件将总条数返回。

```
POST https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/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" ]
  } ]
}
```

响应示例

状态码： 200

- 查询操作成功。
- 示例1：action为filter时的响应体
- 示例2：action为count时的响应体
- 示例 1

```
{
  "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
}
```

- 示例 2

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

SDK 代码示例

SDK代码示例如下。

Java

- 查询公网NAT网关资源实例，其中，操作标识为filter，进行分页查询，查询记录数为100条。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListNatGatewayByTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListNatGatewayByTagRequest request = new ListNatGatewayByTagRequest();
        ListNatsByTagsRequestBody body = new ListNatsByTagsRequestBody();
        List<PublicMatch> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new PublicMatch()
                .withKey("resource_name")
                .withValue("nat_gateways")
        );
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("*value7");
        listNotTagsAnyValues.add("value8");
        List<PublicTag> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new PublicTag()
                .withKey("key4")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value1");
        listNotTagsValues.add("value2");
        List<PublicTag> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new PublicTag()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("*value5");
        listTagsAnyValues.add("value6");
        List<PublicTag> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
```

```
            new PublicTag()
                .withKey("key3")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value3");
        listTagsValues.add("value4");
        List<PublicTag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new PublicTag()
                .withKey("key2")
                .withValues(listTagsValues)
        );
        body.withMatches(listbodyMatches);
        body.withAction("filter");
        body.withOffset("100");
        body.withLimit("100");
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withNotTags(listbodyNotTags);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            ListNatGatewayByTagResponse response = client.listNatGatewayByTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 查询公网NAT网关资源实例，其中，操作标识为count，按照条件将总条数返回。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListNatGatewayByTagSolution {

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

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



```
NatClient client = NatClient.newBuilder()
        .withCredential(auth)
        .withRegion(NatRegion.valueOf("<YOUR REGION>"))
        .build();
ListNatGatewayByTagRequest request = new ListNatGatewayByTagRequest();
ListNatsByTagsRequestBody body = new ListNatsByTagsRequestBody();
List<PublicMatch> listbodyMatches = new ArrayList<>();
listbodyMatches.add(
    new PublicMatch()
        .withKey("resource_name")
        .withValue("nat_gateways")
);
List<String> listNotTagsAnyValues = new ArrayList<>();
listNotTagsAnyValues.add("*value7");
listNotTagsAnyValues.add("value8");
List<PublicTag> listbodyNotTagsAny = new ArrayList<>();
listbodyNotTagsAny.add(
    new PublicTag()
        .withKey("key4")
        .withValues(listNotTagsAnyValues)
);
List<String> listNotTagsValues = new ArrayList<>();
listNotTagsValues.add("*value1");
listNotTagsValues.add("value2");
List<PublicTag> listbodyNotTags = new ArrayList<>();
listbodyNotTags.add(
    new PublicTag()
        .withKey("key1")
        .withValues(listNotTagsValues)
);
List<String> listTagsAnyValues = new ArrayList<>();
listTagsAnyValues.add("*value5");
listTagsAnyValues.add("value6");
List<PublicTag> listbodyTagsAny = new ArrayList<>();
listbodyTagsAny.add(
    new PublicTag()
        .withKey("key3")
        .withValues(listTagsAnyValues)
);
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("*value3");
listTagsValues.add("value4");
List<PublicTag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new PublicTag()
        .withKey("key2")
        .withValues(listTagsValues)
);
body.withMatches(listbodyMatches);
body.withAction("count");
body.withNotTagsAny(listbodyNotTagsAny);
body.withNotTags(listbodyNotTags);
body.withTagsAny(listbodyTagsAny);
body.withTags(listbodyTags);
request.withBody(body);
try {
    ListNatGatewayByTagResponse response = client.listNatGatewayByTag(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
```

```
    }
}
```

Python

- 查询公网NAT网关资源实例，其中，操作标识为filter，进行分页查询，查询记录数为100条。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListNatGatewayByTagRequest()
        listMatchesbody = [
            PublicMatch(
                key="resource_name",
                value="nat_gateways"
            )
        ]
        listValuesNotTagsAny = [
            "*value7",
            "value8"
        ]
        listNotTagsAnybody = [
            PublicTag(
                key="key4",
                values=listValuesNotTagsAny
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            PublicTag(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "*value5",
            "value6"
        ]
        listTagsAnybody = [
            PublicTag(
                key="key3",
                values=listValuesTagsAny
            )
```

```
        ]
        listValuesTags = [
            "*value3",
            "value4"
        ]
        listTagsbody = [
            PublicTag(
                key="key2",
                values=listValuesTags
            )
        ]
        request.body = ListNatsByTagsRequestBody(
            matches=listMatchesbody,
            action="filter",
            offset="100",
            limit="100",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        response = client.list_nat_gateway_by_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 查询公网NAT网关资源实例，其中，操作标识为count，按照条件将总条数返回。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListNatGatewayByTagRequest()
        listMatchesbody = [
            PublicMatch(
                key="resource_name",
                value="nat_gateways"
            )
        ]
        listValuesNotTagsAny = [
            "*value7",
            "value8"
        ]
        listNotTagsAnybody = [
            PublicTag(
                key="key4",
                values=listValuesNotTagsAny
```

```
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            PublicTag(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listValuesTagsAny = [
            "*value5",
            "value6"
        ]
        listTagsAnybody = [
            PublicTag(
                key="key3",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value3",
            "value4"
        ]
        listTagsbody = [
            PublicTag(
                key="key2",
                values=listValuesTags
            )
        ]
        request.body = ListNatsByTagsRequestBody(
            matches=listMatchesbody,
            action="count",
            not_tags_any=listNotTagsAnybody,
            not_tags=listNotTagsbody,
            tags_any=listTagsAnybody,
            tags=listTagsbody
        )
        response = client.list_nat_gateway_by_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 查询公网NAT网关资源实例，其中，操作标识为filter，进行分页查询，查询记录数为100条。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNatGatewayByTagRequest{}
    var listMatchesbody = []model.PublicMatch{
        {
            Key: "resource_name",
            Value: "nat_gateways",
        },
    }
    var listValuesNotTagsAny = []string{
        "*value7",
        "value8",
    }
    var listNotTagsAnybody = []model.PublicTag{
        {
            Key: "key4",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.PublicTag{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listValuesTagsAny = []string{
        "*value5",
        "value6",
    }
    var listTagsAnybody = []model.PublicTag{
        {
            Key: "key3",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value3",
        "value4",
    }
    var listTagsbody = []model.PublicTag{
        {
            Key: "key2",
            Values: listValuesTags,
        },
    }
    offsetListNatsByTagsRequestBody:= "100"
    limitListNatsByTagsRequestBody:= "100"
    request.Body = &model.ListNatsByTagsRequestBody{
        Matches: &listMatchesbody,
        Action: "filter",
        Offset: &offsetListNatsByTagsRequestBody,
        Limit: &limitListNatsByTagsRequestBody,
```

```
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    response, err := client.ListNatGatewayByTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 查询公网NAT网关资源实例，其中，操作标识为count，按照条件将总条数返回。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListNatGatewayByTagRequest{}
    var listMatchesbody = []model.PublicMatch{
        {
            Key: "resource_name",
            Value: "nat_gateways",
        },
    }
    var listValuesNotTagsAny = []string{
        "*value7",
        "value8",
    }
    var listNotTagsAnybody = []model.PublicTag{
        {
            Key: "key4",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.PublicTag{
        {
            Key: "key1",
            Values: listValuesNotTags,
```

```
        },
    }
    var listValuesTagsAny = []string{
        "*value5",
        "value6",
    }
    var listTagsAnybody = []model.PublicTag{
        {
            Key: "key3",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value3",
        "value4",
    }
    var listTagsbody = []model.PublicTag{
        {
            Key: "key2",
            Values: listValuesTags,
        },
    }
    request.Body = &model.ListNatsByTagsRequestBody{
        Matches: &listMatchesbody,
        Action: "count",
        NotTagsAny: &listNotTagsAnybody,
        NotTags: &listNotTagsbody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
    }
    response, err := client.ListNatGatewayByTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	• 查询操作成功。 • 示例1：action为filter时的响应体 • 示例2：action为count时的响应体

错误码

请参见**错误码**。

4.5.2 批量添加/删除公网 NAT 网关资源标签

功能介绍

- 为指定公网NAT网关实例批量添加或删除标签。
- 标签管理服务需要使用该接口批量管理实例的标签。
- 一个资源上最多有10个标签。

接口约束

- 此接口为幂等接口：
  - 创建时如果请求体中存在重复key则报错。
  - 创建时，不允许重复key，如果数据库存在就覆盖。
  - 删除时，如果删除的标签不存在，默认处理成功，删除时不对标签字符集范围做校验。key最大长度为128个unicode字符，value最大长度为255个unicode字符。删除时tags结构体不能缺失，key不能为空，或者空字符串。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/nat_gateways/{nat_gateway_id}/tags/action

表 4-116 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	公网NAT网关ID。

请求参数

表 4-117 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-118 请求 Body 参数

参数	是否必选	参数类型	描述
tags	是	Array of PublicTags objects	标签列表。请参考表Tags字段数据结构说明
action	是	String	操作标识：仅限于create（创建）、delete（删除）

表 4-119 PublicTags

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	是	String	值。每个值最大长度255个unicode字符，删除时如果value有值按照key/value删除，如果value没值，则按照key删除，可以为空字符串。,不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

响应参数

无

请求示例

- 批量添加公网NAT网关资源标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
POST https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

- 批量删除公网NAT网关资源标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
{
  "action" : "delete",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

- 批量添加公网NAT网关资源标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeleteNatGatewayTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateDeleteNatGatewayTagRequest request = new BatchCreateDeleteNatGatewayTagRequest();
        BatchCreateDeleteNatTagsRequestBody body = new BatchCreateDeleteNatTagsRequestBody();
```

```
        List<PublicTags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new PublicTags()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new PublicTags()
                .withKey("key2")
                .withValue("value2")
        );
        body.withAction("create");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateDeleteNatGatewayTagResponse response =
client.batchCreateDeleteNatGatewayTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 批量删除公网NAT网关资源标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeleteNatGatewayTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
```

```
        BatchCreateDeleteNatGatewayTagRequest request = new
BatchCreateDeleteNatGatewayTagRequest();
        BatchCreateDeleteNatTagsRequestBody body = new BatchCreateDeleteNatTagsRequestBody();
        List<PublicTags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new PublicTags()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new PublicTags()
                .withKey("key2")
                .withValue("value2")
        );
        body.withAction("delete");
        body.withTags(listbodyTags);
        request.withBody(body);
        try {
            BatchCreateDeleteNatGatewayTagResponse response =
client.batchCreateDeleteNatGatewayTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 批量添加公网NAT网关资源标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeleteNatGatewayTagRequest()
        listTagsbody = [
            PublicTags(
```

```
            key="key1",
            value="value1"
        ),
        PublicTags(
            key="key2",
            value="value2"
        )
    ]
    request.body = BatchCreateDeleteNatTagsRequestBody(
        action="create",
        tags=listTagsbody
    )
    response = client.batch_create_delete_nat_gateway_tag(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 批量删除公网NAT网关资源标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeleteNatGatewayTagRequest()
        listTagsbody = [
            PublicTags(
                key="key1",
                value="value1"
            ),
            PublicTags(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchCreateDeleteNatTagsRequestBody(
            action="delete",
            tags=listTagsbody
        )
        response = client.batch_create_delete_nat_gateway_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

- 批量添加公网NAT网关资源标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeleteNatGatewayTagRequest{}
    var listTagsbody = []model.PublicTags{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchCreateDeleteNatTagsRequestBody{
        Action: "create",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateDeleteNatGatewayTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 批量删除公网NAT网关资源标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeleteNatGatewayTagRequest{}
    var listTagsbody = []model.PublicTags{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchCreateDeleteNatTagsRequestBody{
        Action: "delete",
        Tags: listTagsbody,
    }
    response, err := client.BatchCreateDeleteNatGatewayTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	批量添加或删除操作成功。

错误码

请参见**错误码**。

4.5.3 添加公网 NAT 网关资源标签

功能介绍

- 添加公网NAT网关资源标签。一个资源上最多有10个标签。

接口约束

- 此接口为幂等接口：
  - 创建时，如果创建的标签已经存在（key相同），则覆盖。
- 创建tag时，要求网关存在。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/nat_gateways/{nat_gateway_id}/tags

表 4-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	所属公网NAT网关的id。

请求参数

表 4-121 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

表 4-122 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	TagBody object	标签列表。请参考表TagBody字段数据结构说明。

表 4-123 TagBody

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	是	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

响应参数

无

请求示例

添加公网NAT网关资源标签，其中，标签键为“key1”，对应的值为“value1”。

```
POST  https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags

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

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

添加公网NAT网关资源标签，其中，标签键为“key1”，对应的值为“value1”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreateNatGatewayTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateNatGatewayTagRequest request = new CreateNatGatewayTagRequest();
        CreateNatTagRequestBody body = new CreateNatTagRequestBody();
        TagBody tagbody = new TagBody();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateNatGatewayTagResponse response = client.createNatGatewayTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

添加公网NAT网关资源标签，其中，标签键为“key1”，对应的值为“value1”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateNatGatewayTagRequest()
        tagbody = TagBody(
            key="key1",
            value="value1"
        )
        request.body = CreateNatTagRequestBody(
            tag=tagbody
        )
        response = client.create_nat_gateway_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

添加公网NAT网关资源标签，其中，标签键为“key1”，对应的值为“value1”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateNatGatewayTagRequest{}
    tagbody := &model.TagBody{
        Key: "key1",
        Value: "value1",
    }
```

```
    request.Body = &model.CreateNatTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateNatGatewayTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	添加操作成功。

错误码

请参见**错误码**。

4.5.4 删除公网 NAT 网关资源标签

功能介绍

- 删除指定公网NAT网关资源实例的标签信息。

接口约束

- 此接口为幂等接口：
  - 删除时，不对标签做校验。删除的key不存在报404，key不能为空或者空字符串。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/nat_gateways/{nat_gateway_id}/tags/{key}

表 4-124 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	公网NAT网关id。
key	是	String	标签key。

请求参数

表 4-125 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

无

请求示例

```
DELETE  https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags/key1
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeleteNatGatewayTagSolution {
```

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteNatGatewayTagRequest request = new DeleteNatGatewayTagRequest();
        try {
            DeleteNatGatewayTagResponse response = client.deleteNatGatewayTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除操作成功。

错误码

请参见**错误码**。

4.5.5 查询公网 NAT 网关资源标签

功能介绍

- 查询指定公网NAT网关实例的标签信息。
- 标签管理服务需要使用该接口查询指定公网NAT网关实例的全部标签数据。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/nat_gateways/{nat_gateway_id}/tags

表 4-126 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。
nat_gateway_id	是	String	公网NAT网关ID。

请求参数

表 4-127 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

状态码： 200

表 4-128 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagBody** objects	标签列表。

表 4-129 TagBody

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”
value	String	值。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，“=”，“*”，“<”，“>”，“\”，“,”，“|”，“/”

请求示例

```
GET https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/fe1a4cf0-27fe-4b97-a9b1-2c67c127f0e0/tags
```

响应示例

状态码： 200

查询操作成功。

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

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;
```

```
public class ShowNatGatewayTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowNatGatewayTagRequest request = new ShowNatGatewayTagRequest();
        try {
            ShowNatGatewayTagResponse response = client.showNatGatewayTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowNatGatewayTagRequest()
        response = client.show_nat_gateway_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

4.5.6 查询公网 NAT 网关项目标签

功能介绍

- 查询租户在指定项目和公网NAT网关实例类型的所有标签集合。
- 标签管理服务需要能够列出当前租户全部已使用的标签集合，为各服务Console打标签和过滤实例时提供标签联想功能。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/nat_gateways/tags

表 4-130 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。

请求参数

表 4-131 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。

响应参数

状态码： 200

表 4-132 响应 Body 参数

参数	参数类型	描述
tags	Array of **TagsBody** objects	标签列表。

表 4-133 TagsBody

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。不能包含非打印字符ASCII(0-31)，*,<,>,\,=
values	Array of strings	值列表。每个值最大长度255个unicode字符，可以为空字符串。不能包含非打印字符ASCII(0-31)，*,<,>,\,=

请求示例

```
GET https://{Endpoint}/v3/9ad601814ac94c80bf7bb9073ded66fc/nat_gateways/tags
```

响应示例

状态码： 200

查询操作成功。

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

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

5 API（私网 NAT 网关）

5.1 私网 NAT 网关

5.1.1 查询私网 NAT 网关列表

功能介绍

查询私网NAT网关实例列表。

接口约束

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。 支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/gateways

表 5-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 5-2 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
marker	否	String	功能说明：分页查询起始的资源ID，为空时查询第一页。 值从上一次查询的PageInfo中的next_marker或者previous_marker中获取。 最小长度：**36** 最大长度：**36**
page_reverse	否	Boolean	是否查询前一页。
id	否	Array	私网NAT网关实例的ID。 数组长度：**1 - 10**
name	否	Array	私网NAT网关实例的名字。 数组长度：**1 - 10**
description	否	Array	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 数组长度：**1 - 10**
spec	否	Array	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 数组长度：**1 - 10** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**
project_id	否	Array	项目的ID。 数组长度：**1 - 10**

参数	是否必选	参数类型	描述
status	否	Array	私网NAT网关实例的状态。 取值为： "ACTIVE"：正常运行 "FROZEN"：冻结 数组长度：**1 - 10** 枚举值： • **ACTIVE** • **FROZEN**
vpc_id	否	Array	私网NAT网关实例所属VPC的ID。 数组长度：**1 - 10**
virsubnet_id	否	Array	私网NAT网关实例所属子网的ID。 数组长度：**1 - 10**
enterprise_project_id	否	Array	企业项目ID。创建私网NAT网关实例时，关联的企业项目ID。 数组长度：**1 - 10**

请求参数

表 5-3 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-4 响应 Body 参数

参数	参数类型	描述
gateways	Array of **PrivateNat** objects	查询私网NAT网关实例列表的响应体。 详见PrivateNat字段说明。 数组长度：**0 - 2000**
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**
page_info	**PageInfo** object	分页信息。

表 5-5 PrivateNat

参数	参数类型	描述
id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**32** 最大长度：**32**
name	String	私网NAT网关实例的名字。 最小长度：**1** 最大长度：**64**
description	String	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 缺省值：**Small** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**

参数	参数类型	描述
status	String	私网NAT网关实例的状态。 取值为： "ACTIVE"：正常运行 "FROZEN"：冻结 枚举值： • **ACTIVE** • **FROZEN**
created_at	String	私网NAT网关实例的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
updated_at	String	私网NAT网关实例的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
downlink_vpcs	Array of **DownlinkVpc** objects	私网NAT网关实例所属的VPC实例。 数组长度：**1 - 10**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
enterprise_project_id	String	企业项目ID。 创建私网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-6 DownlinkVpc

参数	参数类型	描述
vpc_id	String	私网NAT网关实例所属VPC的ID。 最小长度：**36** 最大长度：**36**
virsubnet_id	String	私网NAT网关实例所属子网的ID。 最小长度：**36** 最大长度：**36**

表 5-7 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**

参数	参数类型	描述
value	String	标签value。 最小长度：**0** 最大长度：**255**

表 5-8 PageInfo

参数	参数类型	描述
next_marker	String	分页查询结果中最后一条记录的ID。通常用于查询下一页。 最小长度：**1** 最大长度：**36**
previous_marker	String	分页查询结果中第一条记录的ID。通常用于配合page_reverse=true查询上一页。 最小长度：**1** 最大长度：**36**
current_count	Integer	分页查询资源时，本页的实例的个数。 最小值：**1** 最大值：**2000**

请求示例

```
GET https://{Endpoint}/v3/70505c941b9b4dfd82fd351932328a2f/private-nat/gateways
```

响应示例

状态码： 200

查询私网NAT网关实例列表成功

```
{
  "gateways" : [ {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "private-nat-gateway-name1",
    "description" : "private-nat-gateway-description1",
    "spec" : "Small",
    "project_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22T08:47:13",
    "updated_at" : "2019-04-22T08:47:13",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ],
    "downlink_vpcs" : [ {
      "vpc_id" : "3cb66d44-9f75-4237-bfff-e37b14d23ad2",
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
    } ]
  }, {
```

```
    "id" : "65995b8e-dcb7-4ab4-9931-bc3c95beec0a",
    "name" : "private-nat-gateway-name2",
    "description" : "private-nat-gateway-description2",
    "spec" : "Small",
    "project_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22T09:06:54",
    "updated_at" : "2019-04-22T09:06:54",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ],
    "downlink_vpcs" : [ {
      "vpc_id" : "3cb66d44-9f75-4237-bfff-e37b14d23ad2",
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
    } ]
  } ],
  "request_id" : "a7b00469-5a31-4274-bb10-59167243383e",
  "page_info" : {
    "previous_marker" : "14338426-6afe-4019-996b-3a9525296e11",
    "current_count" : 2
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListPrivateNatsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPrivateNatsRequest request = new ListPrivateNatsRequest();
        request.withLimit(<limit>);
```

```
        request.withMarker("<marker>");
        request.withPageReverse(<page_reverse>);
        request.withId();
        request.withName();
        request.withDescription();
        request.withSpec();
        request.withStatus();
        request.withVpcId();
        request.withVirsubnetId();
        request.withEnterpriseProjectId();
        try {
            ListPrivateNatsResponse response = client.listPrivateNats(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ListPrivateNatsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.page_reverse = <PageReverse>
        request.id =
        request.name =
        request.description =
        request.spec =
        request.status =
        request.vpc_id =
        request.virsubnet_id =
        request.enterprise_project_id =
        response = client.list_private_nats(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPrivateNatsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    pageReverseRequest:= <page_reverse>
    request.PageReverse = &pageReverseRequest
    response, err := client.ListPrivateNats(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询私网NAT网关实例列表成功

错误码

请参见**错误码**。

5.1.2 更新私网 NAT 网关

功能介绍

更新私网NAT网关实例。

调用方法

请参见**如何调用API**。

URI

PUT /v3/{project_id}/private-nat/gateways/{gateway_id}

表 5-9 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
gateway_id	是	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-10 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-11 请求 Body 参数

参数	是否必选	参数类型	描述
gateway	是	**UpdatePrivateNatOption** object	更新私网NAT网关实例的请求体。

表 5-12 UpdatePrivateNatOption

参数	是否必选	参数类型	描述
name	否	String	私网NAT网关实例的名字。 私网NAT网关实例的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。 最小长度：**1** 最大长度：**64**
description	否	String	私网NAT网关的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	否	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**

响应参数

状态码： 200

表 5-13 响应 Body 参数

参数	参数类型	描述
gateway	**PrivateNat** object	私网NAT网关实例的响应体。

参数	参数类型	描述
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-14 PrivateNat

参数	参数类型	描述
id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**32** 最大长度：**32**
name	String	私网NAT网关实例的名字。 最小长度：**1** 最大长度：**64**
description	String	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 缺省值：**Small** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**
status	String	私网NAT网关实例的状态。 取值为： "ACTIVE"：正常运行 "FROZEN"：冻结 枚举值： • **ACTIVE** • **FROZEN**
created_at	String	私网NAT网关实例的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。

参数	参数类型	描述
updated_at	String	私网NAT网关实例的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
downlink_vpcs	Array of **DownlinkVpc** objects	私网NAT网关实例所属的VPC实例。 数组长度：**1 - 10**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
enterprise_project_id	String	企业项目ID。 创建私网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-15 DownlinkVpc

参数	参数类型	描述
vpc_id	String	私网NAT网关实例所属VPC的ID。 最小长度：**36** 最大长度：**36**
virsubnet_id	String	私网NAT网关实例所属子网的ID。 最小长度：**36** 最大长度：**36**

表 5-16 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

更新私网NAT网关实例，其中，私网NAT网关实例的名字为private-nat-gateway-name，私网NAT网关的描述为private-nat-gateway-description，实例的规格为Medium。

```
PUT https://{Endpoint}/v3/70505c941b9b4dfd82fd351932328a2f/private-nat/gateways/
14338426-6afe-4019-996b-3a9525296e11

{
  "gateway" : {
    "name" : "private-nat-gateway-name",
    "description" : "private-nat-gateway-description",
    "spec" : "Medium"
  }
}
```

响应示例

状态码： 200

更新私网NAT网关实例成功。

```
{
  "gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "private-nat-gateway-name",
    "description" : "private-nat-gateway-description",
    "spec" : "Medium",
    "project_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22T08:47:13",
    "updated_at" : "2019-04-22T08:47:13",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ],
    "downlink_vpcs" : [ {
      "vpc_id" : "3cb66d44-9f75-4237-bfff-e37b14d23ad2",
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
    } ]
  },
  "request_id" : "e7e3323e95b348708d26e68a0ddece71"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新私网NAT网关实例，其中，私网NAT网关实例的名字为private-nat-gateway-name，私网NAT网关的描述为private-nat-gateway-description，实例的规格为Medium。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class UpdatePrivateNatSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```



```
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePrivateNatRequest request = new UpdatePrivateNatRequest();
        UpdatePrivateNatRequestBody body = new UpdatePrivateNatRequestBody();
        UpdatePrivateNatOption gatewaybody = new UpdatePrivateNatOption();
        gatewaybody.withName("private-nat-gateway-name")
            .withDescription("private-nat-gateway-description")
            .withSpec(UpdatePrivateNatOption.SpecEnum.fromValue("Medium"));
        body.withGateway(gatewaybody);
        request.withBody(body);
        try {
            UpdatePrivateNatResponse response = client.updatePrivateNat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

更新私网NAT网关实例，其中，私网NAT网关实例的名字为private-nat-gateway-name，私网NAT网关的描述为private-nat-gateway-description，实例的规格为Medium。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

    client = NatClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(NatRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
    try:
        request = UpdatePrivateNatRequest()
        gatewaybody = UpdatePrivateNatOption(
            name="private-nat-gateway-name",
            description="private-nat-gateway-description",
            spec="Medium"
        )
        request.body = UpdatePrivateNatRequestBody(
            gateway=gatewaybody
        )
        response = client.update_private_nat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新私网NAT网关实例，其中，私网NAT网关实例的名字为private-nat-gateway-name，私网NAT网关的描述为private-nat-gateway-description，实例的规格为Medium。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePrivateNatRequest{}
    nameGateway:= "private-nat-gateway-name"
    descriptionGateway:= "private-nat-gateway-description"
    specGateway:= model.GetUpdatePrivateNatOptionSpecEnum().MEDIUM
    gatewaybody := &model.UpdatePrivateNatOption{
        Name: &nameGateway,
        Description: &descriptionGateway,
        Spec: &specGateway,
    }
    request.Body = &model.UpdatePrivateNatRequestBody{
        Gateway: gatewaybody,
    }
    response, err := client.UpdatePrivateNat(request)
```

```
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新私网NAT网关实例成功。

错误码

请参见**错误码**。

5.1.3 删除私网 NAT 网关

功能介绍

删除私网NAT网关实例。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/private-nat/gateways/{gateway_id}

表 5-17 路径参数

参数	是否必选	参数类型	描述
gateway_id	是	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-18 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v3/70505c941b9b4dfd82fd351932328a2f/private-nat/gateways/
14338426-6afe-4019-996b-3a9525296e11
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeletePrivateNatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePrivateNatRequest request = new DeletePrivateNatRequest();
        try {
            DeletePrivateNatResponse response = client.deletePrivateNat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除私网NAT网关实例成功。

错误码

请参见**错误码**。

5.1.4 创建私网 NAT 网关

功能介绍

创建私网NAT网关实例。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat/gateways

表 5-19 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-20 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-21 请求 Body 参数

参数	是否必选	参数类型	描述
gateway	是	**CreatePrivateNatOption** object	创建私网NAT网关实例的请求体。

表 5-22 CreatePrivateNatOption

参数	是否必选	参数类型	描述
name	是	String	私网NAT网关实例的名字。 私网NAT网关实例的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。 最小长度：**1** 最大长度：**64**

参数	是否必选	参数类型	描述
description	否	String	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	否	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 缺省值：**Small** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**
downlink_vpcs	是	Array of **DownlinkVpcOption** objects	私网NAT网关实例所属的VPC实例。 数组长度：**1 - 1**
tags	否	Array of **Tag** objects	标签列表 数组长度：**0 - 10**
enterprise_project_id	否	String	企业项目ID 创建私网NAT网关实例时，关联的企业项目ID。关于企业项目ID的获取及企业项目特性的详细信息，请参考《企业管理用户指南》。 缺省值：**0** 最小长度：**1** 最大长度：**36**

表 5-23 DownlinkVpcOption

参数	是否必选	参数类型	描述
virsubnet_id	是	String	私网NAT网关实例所属的子网的ID。 最小长度：**36** 最大长度：**36**

表 5-24 Tag

参数	是否必选	参数类型	描述
key	是	String	标签key值。 最小长度：**1** 最大长度：**128**
value	是	String	标签value。 最小长度：**0** 最大长度：**255**

响应参数

状态码： 201

表 5-25 响应 Body 参数

参数	参数类型	描述
gateway	**PrivateNat** object	私网NAT网关实例的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-26 PrivateNat

参数	参数类型	描述
id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**32** 最大长度：**32**
name	String	私网NAT网关实例的名字。 最小长度：**1** 最大长度：**64**
description	String	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	参数类型	描述
spec	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 缺省值：**Small** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**
status	String	私网NAT网关实例的状态。 取值为： "ACTIVE"：正常运行 "FROZEN"：冻结 枚举值： • **ACTIVE** • **FROZEN**
created_at	String	私网NAT网关实例的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
updated_at	String	私网NAT网关实例的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
downlink_vpcs	Array of **DownlinkVpc** objects	私网NAT网关实例所属的VPC实例。 数组长度：**1 - 10**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
enterprise_project_id	String	企业项目ID。 创建私网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-27 DownlinkVpc

参数	参数类型	描述
vpc_id	String	私网NAT网关实例所属VPC的ID。 最小长度：**36** 最大长度：**36**
virsubnet_id	String	私网NAT网关实例所属子网的ID。 最小长度：**36** 最大长度：**36**

表 5-28 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

创建私网NAT网关实例，其中，私网NAT网关实例的名称为private-nat-gateway-name，实例的规格为Small，所属的子网的id为373979ee-f4f0-46c5-80e3-0fbf72646b70。

```
POST https://{Endpoint}/v3/70505c941b9b4dfd82fd351932328a2f/private-nat/gateways

{
  "gateway" : {
    "name" : "private-nat-gateway-name",
    "description" : "private-nat-gateway-description",
    "spec" : "Small",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "downlink_vpcs" : [ {
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
    } ],
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ]
  }
}
```

响应示例

状态码： 201

创建私网NAT网关实例成功。

```
{
  "request_id" : "9882046a9b96f1405472e36d797e33dc",
  "gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "private-nat-gateway-name",
    "description" : "private-nat-gateway-description",
    "spec" : "Small",
    "project_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22T08:47:13",
    "updated_at" : "2019-04-22T08:47:13",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ],
    "downlink_vpcs" : [ {
      "vpc_id" : "3cb66d44-9f75-4237-bfff-e37b14d23ad2",
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
```

```
  }]
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建私网NAT网关实例，其中，私网NAT网关实例的名称为private-nat-gateway-name，实例的规格为Small，所属的子网的id为373979ee-f4f0-46c5-80e3-0fbf72646b70。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class CreatePrivateNatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePrivateNatRequest request = new CreatePrivateNatRequest();
        CreatePrivateNatRequestBody body = new CreatePrivateNatRequestBody();
        List<PrivateTag> listGatewayTags = new ArrayList<>();
        listGatewayTags.add(
            new PrivateTag()
                .withKey("key1")
                .withValue("value1")
        );
        List<DownlinkVpcOption> listGatewayDownlinkVpcs = new ArrayList<>();
        listGatewayDownlinkVpcs.add(
            new DownlinkVpcOption()
                .withVirsubnetId("373979ee-f4f0-46c5-80e3-0fbf72646b70")
        );
        CreatePrivateNatOption gatewaybody = new CreatePrivateNatOption();
        gatewaybody.withName("private-nat-gateway-name")
            .withDescription("private-nat-gateway-description")
            .withSpec(CreatePrivateNatOption.SpecEnum.fromValue("Small"))
            .withDownlinkVpcs(listGatewayDownlinkVpcs)
            .withTags(listGatewayTags)
            .withEnterpriseProjectId("2759da7b-8015-404c-ae0a-a389007b0e2a");
```

```
        body.withGateway(gatewaybody);
        request.withBody(body);
        try {
            CreatePrivateNatResponse response = client.createPrivateNat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建私网NAT网关实例，其中，私网NAT网关实例的名称为private-nat-gateway-name，实例的规格为Small，所属的子网的id为373979ee-f4f0-46c5-80e3-0fbf72646b70。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreatePrivateNatRequest()
        listTagsGateway = [
            PrivateTag(
                key="key1",
                value="value1"
            )
        ]
        listDownlinkVpcsGateway = [
            DownlinkVpcOption(
                virsubnet_id="373979ee-f4f0-46c5-80e3-0fbf72646b70"
            )
        ]
        gatewaybody = CreatePrivateNatOption(
            name="private-nat-gateway-name",
            description="private-nat-gateway-description",
            spec="Small",
            downlink_vpcs=listDownlinkVpcsGateway,
            tags=listTagsGateway,
            enterprise_project_id="2759da7b-8015-404c-ae0a-a389007b0e2a"
```

```
        )
        request.body = CreatePrivateNatRequestBody(
            gateway=gatewaybody
        )
        response = client.create_private_nat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建私网NAT网关实例，其中，私网NAT网关实例的名称为private-nat-gateway-name，实例的规格为Small，所属的子网的id为373979ee-f4f0-46c5-80e3-0fbf72646b70。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePrivateNatRequest{}
    var listTagsGateway = []model.PrivateTag{
        {
            Key: "key1",
            Value: "value1",
        },
    }
    var listDownlinkVpcsGateway = []model.DownlinkVpcOption{
        {
            VirsubnetId: "373979ee-f4f0-46c5-80e3-0fbf72646b70",
        },
    }
    descriptionGateway:= "private-nat-gateway-description"
    specGateway:= model.GetCreatePrivateNatOptionSpecEnum().SMALL
    enterpriseProjectIdGateway:= "2759da7b-8015-404c-ae0a-a389007b0e2a"
    gatewaybody := &model.CreatePrivateNatOption{
        Name: "private-nat-gateway-name",
        Description: &descriptionGateway,
        Spec: &specGateway,
        DownlinkVpcs: listDownlinkVpcsGateway,
```

```
        Tags: &listTagsGateway,
        EnterpriseProjectId: &enterpriseProjectIdGateway,
    }
    request.Body = &model.CreatePrivateNatRequestBody{
        Gateway: gatewaybody,
    }
    response, err := client.CreatePrivateNat(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建私网NAT网关实例成功。

错误码

请参见**错误码**。

5.1.5 查询指定的私网 NAT 网关详情

功能介绍

查询指定的私网NAT网关实例详情。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/gateways/{gateway_id}

表 5-29 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

参数	是否必选	参数类型	描述
gateway_id	是	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-30 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-31 响应 Body 参数

参数	参数类型	描述
gateway	**PrivateNat** object	私网NAT网关实例的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-32 PrivateNat

参数	参数类型	描述
id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
project_id	String	项目的ID。 最小长度：**32** 最大长度：**32**
name	String	私网NAT网关实例的名字。 最小长度：**1** 最大长度：**64**
description	String	私网NAT网关实例的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
spec	String	私网NAT网关实例的规格。 取值为： "Small"：小型 "Medium"：中型 "Large"：大型 "Extra-large"：超大型 缺省值：**Small** 枚举值： • **Small** • **Medium** • **Large** • **Extra-large**
status	String	私网NAT网关实例的状态。 取值为： "ACTIVE"：正常运行 "FROZEN"：冻结 枚举值： • **ACTIVE** • **FROZEN**
created_at	String	私网NAT网关实例的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
updated_at	String	私网NAT网关实例的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。
downlink_vpcs	Array of **DownlinkVpc** objects	私网NAT网关实例所属的VPC实例。 数组长度：**1 - 10**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
enterprise_project_id	String	企业项目ID。 创建私网NAT网关实例时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-33 DownlinkVpc

参数	参数类型	描述
vpc_id	String	私网NAT网关实例所属VPC的ID。 最小长度：**36** 最大长度：**36**
virsubnet_id	String	私网NAT网关实例所属子网的ID。 最小长度：**36** 最大长度：**36**

表 5-34 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/70505c941b9b4dfd82fd351932328a2f/private-nat/gateways/
14338426-6afe-4019-996b-3a9525296e11
```

响应示例

状态码： 200

查询私网NAT网关实例成功。

```
{
  "gateway" : {
    "id" : "14338426-6afe-4019-996b-3a9525296e11",
    "name" : "private-nat-gateway-name",
    "description" : "private-nat-gateway-description",
    "spec" : "Small",
    "project_id" : "70505c941b9b4dfd82fd351932328a2f",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "status" : "ACTIVE",
    "created_at" : "2019-04-22T08:47:13",
    "updated_at" : "2019-04-22T08:47:13",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ],
    "downlink_vpcs" : [ {
      "vpc_id" : "3cb66d44-9f75-4237-bfff-e37b14d23ad2",
      "virsubnet_id" : "373979ee-f4f0-46c5-80e3-0fbf72646b70"
    } ]
  },
```

```
  "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ShowPrivateNatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPrivateNatRequest request = new ShowPrivateNatRequest();
        try {
            ShowPrivateNatResponse response = client.showPrivateNat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *
```

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询私网NAT网关实例成功。

错误码

请参见**错误码**。

5.2 DNAT 规则

5.2.1 查询 DNAT 规则列表

功能介绍

查询DNAT规则列表。

接口约束

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。 支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/dnat-rules

表 5-35 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 5-36 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
marker	否	String	功能说明：分页查询起始的资源ID，为空时查询第一页。 值从上一次查询的PageInfo中的next_marker或者previous_marker中获取。 最小长度：**36** 最大长度：**36**
page_reverse	否	Boolean	是否查询前一页。
id	否	Array	DNAT规则的ID。 数组长度：**1 - 10**
project_id	否	Array	项目的ID。 数组长度：**1 - 10**
enterprise_project_id	否	Array	企业项目ID。创建DNAT规则时，关联的企业项目ID。 数组长度：**1 - 10**
description	否	Array	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 数组长度：**1 - 10**
gateway_id	否	Array	私网NAT网关实例的ID。 数组长度：**1 - 10**
transit_ip_id	否	Array	中转IP的ID。 数组长度：**1 - 10**
external_ip_address	否	Array	中转IP的地址。 数组长度：**1 - 10**
network_interface_id	否	Array	网络接口ID，支持计算、ELB、VIP等实例的网络接口。 数组长度：**1 - 10**

参数	是否必选	参数类型	描述
type	否	Array	DNAT规则后端的类型。 取值：COMPUTE：后端为计算实例。 VIP：后端为VIP的实例。 ELB：后端为ELB的实例。 ELBv3：后端为ELBv3的实例。 CUSTOMIZE：后端为自定义IP。 数组长度：**1 - 10**
private_ip_address	否	Array	后端实例的IP私网地址。 数组长度：**1 - 10**

请求参数

表 5-37 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-38 响应 Body 参数

参数	参数类型	描述
dnat_rules	Array of **PrivateDnat** objects	查询DNAT规则列表的响应体。 数组长度：**0 - 2000**
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**
page_info	**PageInfo** object	分页信息。

表 5-39 PrivateDnat

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	网络接口ID，支持计算、ELB、VIP等实例的端口。 最小长度：**1** 最大长度：**36**
type	String	DNAT规则后端的类型。 取值： COMPUTE：后端为计算实例。 VIP：后端为VIP的实例。 ELB：后端为ELB的实例。 ELBv3：后端为ELBv3的实例。 CUSTOMIZE：后端为自定义IP。 最小长度：**1** 最大长度：**10**
protocol	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**

参数	参数类型	描述
private_ip_address	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
internal_service_port	String	后端实例的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
transit_service_port	String	中转IP的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
enterprise_project_id	String	企业项目ID。创建DNAT规则时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
created_at	String	DNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	DNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**

表 5-40 PageInfo

参数	参数类型	描述
next_marker	String	分页查询结果中最后一条记录的ID。通常用于查询下一页。 最小长度：**1** 最大长度：**36**
previous_marker	String	分页查询结果中第一条记录的ID。通常用于配合page_reverse=true查询上一页。 最小长度：**1** 最大长度：**36**

参数	参数类型	描述
current_count	Integer	分页查询资源时，本页的实例的个数。 最小值：**1** 最大值：**2000**

请求示例

```
GET https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/dnat-rules
```

响应示例

状态码： 200

查询DNAT规则列表成功。

```
{
  "dnat_rules" : [ {
    "id" : "24dd6bf5-48f2-4915-ad0b-5bb111d39c83",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "description" : "aa",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "network_interface_id" : "dae9393a-b536-491c-a5a2-72edc1104707",
    "type" : "COMPUTE",
    "protocol" : "any",
    "internal_service_port" : "0",
    "transit_service_port" : "0",
    "private_ip_address" : "192.168.1.72",
    "created_at" : "2019-04-29T07:10:01",
    "updated_at" : "2019-04-29T07:10:01"
  }, {
    "id" : "25dcdb21-97de-43cd-b476-31637a47f05d",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "description" : "aa",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "15abdf29-4a68-474c-9963-79c4e6d495d7",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "network_interface_id" : "9e2f0dbb-68b2-4c4b-9298-fa4f13187976",
    "type" : "COMPUTE",
    "protocol" : "any",
    "internal_service_port" : "0",
    "transit_service_port" : "0",
    "private_ip_address" : "192.168.1.99",
    "created_at" : "2019-04-29T07:15:41",
    "updated_at" : "2019-04-29T07:15:41"
  } ],
  "request_id" : "a7b00469-5a31-4274-bb10-59167243383e",
  "page_info" : {
    "previous_marker" : "14338426-6afe-4019-996b-018008113013",
    "current_count" : 2
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListPrivateDnatsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPrivateDnatsRequest request = new ListPrivateDnatsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withPageReverse(<page_reverse>);
        request.withId();
        request.withEnterpriseProjectId();
        request.withDescription();
        request.withGatewayId();
        request.withTransitIpId();
        request.withExternalIpAddress();
        request.withNetworkInterfaceId();
        request.withType();
        request.withPrivateIpAddress();
        try {
            ListPrivateDnatsResponse response = client.listPrivateDnats(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ListPrivateDnatsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.page_reverse = <PageReverse>
        request.id =
        request.enterprise_project_id =
        request.description =
        request.gateway_id =
        request.transit_ip_id =
        request.external_ip_address =
        request.network_interface_id =
        request.type =
        request.private_ip_address =
        response = client.list_private_dnats(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPrivateDnatsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    pageReverseRequest:= <page_reverse>
    request.PageReverse = &pageReverseRequest
    response, err := client.ListPrivateDnats(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询DNAT规则列表成功。

错误码

请参见**错误码**。

5.2.2 更新 DNAT 规则

功能介绍

更新指定的DNAT规则。

调用方法

请参见**如何调用API**。

URI

PUT /v3/{project_id}/private-nat/dnat-rules/{dnat_rule_id}

表 5-41 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-42 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-43 请求 Body 参数

参数	是否必选	参数类型	描述
dnat_rule	否	**UpdatePrivateDnatOption** object	更新DNAT规则的请求体。

表 5-44 UpdatePrivateDnatOption

参数	是否必选	参数类型	描述
description	否	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**

参数	是否必选	参数类型	描述
transit_ip_id	否	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
network_interface_id	否	String	网络接口ID，支持计算、ELB、VIP等实例的网络接口。 最小长度：**36** 最大长度：**36**
private_ip_address	否	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
protocol	否	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
internal_service_port	否	String	后端实例的端口号。 最小长度：**1** 最大长度：**5**
transit_service_port	否	String	中转IP的端口号。 最小长度：**1** 最大长度：**10**

响应参数

状态码： 200

表 5-45 响应 Body 参数

参数	参数类型	描述
dnat_rule	**PrivateDnat** object	DNAT规则的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-46 PrivateDnat

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	网络接口ID，支持计算、ELB、VIP等实例的端口。 最小长度：**1** 最大长度：**36**
type	String	DNAT规则后端的类型。 取值： COMPUTE：后端为计算实例。 VIP：后端为VIP的实例。 ELB：后端为ELB的实例。 ELBv3：后端为ELBv3的实例。 CUSTOMIZE：后端为自定义IP。 最小长度：**1** 最大长度：**10**
protocol	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**

参数	参数类型	描述
private_ip_address	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
internal_service_port	String	后端实例的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
transit_service_port	String	中转IP的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
enterprise_project_id	String	企业项目ID。创建DNAT规则时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
created_at	String	DNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	DNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**

请求示例

更新DNAT规则，其中，DNAT规则的描述为my dnat-rules 03。

```
PUT https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/dnat-rules/
24dd6bf5-48f2-4915-ad0b-5bb111d39c83

{
  "dnat_rule" : {
    "description" : "my dnat-rules 03"
  }
}
```

响应示例

状态码： 200

更新DNAT规则成功。

```
{
  "dnat_rule" : {
    "id" : "24dd6bf5-48f2-4915-ad0b-5bb111d39c83",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "description" : "dnat rule description",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
    "network_interface_id" : "dae9393a-b536-491c-a5a2-72edc1104707",
    "type" : "COMPUTE",
    "private_ip_address" : "192.168.1.72",
    "created_at" : "2019-04-29T07:10:01",
    "updated_at" : "2019-04-29T07:10:01"
  },
  "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新DNAT规则，其中，DNAT规则的描述为my dnat-rules 03。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class UpdatePrivateDnatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePrivateDnatRequest request = new UpdatePrivateDnatRequest();
        UpdatePrivateDnatRequestBody body = new UpdatePrivateDnatRequestBody();
        UpdatePrivateDnatOption dnatRulebody = new UpdatePrivateDnatOption();
        dnatRulebody.withDescription("my dnat-rules 03");
        body.withDnatRule(dnatRulebody);
        request.withBody(body);
        try {
            UpdatePrivateDnatResponse response = client.updatePrivateDnat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

更新DNAT规则，其中，DNAT规则的描述为my dnat-rules 03。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdatePrivateDnatRequest()
        dnatRulebody = UpdatePrivateDnatOption(
            description="my dnat-rules 03"
        )
        request.body = UpdatePrivateDnatRequestBody(
            dnat_rule=dnatRulebody
        )
        response = client.update_private_dnat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新DNAT规则，其中，DNAT规则的描述为my dnat-rules 03。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)
```

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePrivateDnatRequest{}
    descriptionDnatRule:= "my dnat-rules 03"
    dnatRulebody := &model.UpdatePrivateDnatOption{
        Description: &descriptionDnatRule,
    }
    request.Body = &model.UpdatePrivateDnatRequestBody{
        DnatRule: dnatRulebody,
    }
    response, err := client.UpdatePrivateDnat(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新DNAT规则成功。

错误码

请参见**错误码**。

5.2.3 创建 DNAT 规则

功能介绍

创建DNAT规则。

接口约束

创建规则时，要求网关状态status = "ACTIVE"。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat/dnat-rules

表 5-47 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-48 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-49 请求 Body 参数

参数	是否必选	参数类型	描述
dnat_rule	是	**CreatePrivateDnatOption** object	创建DNAT规则的请求体。

表 5-50 CreatePrivateDnatOption

参数	是否必选	参数类型	描述
description	否	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
transit_ip_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
network_interface_id	否	String	网络接口ID，支持计算、ELB、VIP等实例的网络接口。 最小长度：**36** 最大长度：**36**
gateway_id	是	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
protocol	否	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
private_ip_address	否	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
internal_service_port	否	String	后端实例的端口号。 最小长度：**1** 最大长度：**5**
transit_service_port	否	String	中转IP的端口号。 最小长度：**1** 最大长度：**5**

响应参数

状态码： 201

表 5-51 响应 Body 参数

参数	参数类型	描述
dnat_rule	**PrivateDnat** object	DNAT规则的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-52 PrivateDnat

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	网络接口ID，支持计算、ELB、VIP等实例的端口。 最小长度：**1** 最大长度：**36**
type	String	DNAT规则后端的类型。 取值： COMPUTE：后端为计算实例。 VIP：后端为VIP的实例。 ELB：后端为ELB的实例。 ELBv3：后端为ELBv3的实例。 CUSTOMIZE：后端为自定义IP。 最小长度：**1** 最大长度：**10**

参数	参数类型	描述
protocol	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**
private_ip_address	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
internal_service_port	String	后端实例的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
transit_service_port	String	中转IP的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
enterprise_project_id	String	企业项目ID。创建DNAT规则时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
created_at	String	DNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	DNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**

请求示例

创建DNAT规则，其中，中转IP的id为3faa719d-6d18-4ccb-a5c7-33e65a09663e，私网NAT网关实例的id为0adefb29-a6c2-48a5-8637-2be67fa03fec。

```
POST https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/dnat-rules

{
  "dnat_rule" : {
    "description" : "aa",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
    "network_interface_id" : "dae9393a-b536-491c-a5a2-72edc1104707"
  }
}
```

响应示例

状态码： 201

创建DNAT规则成功。

```
{
  "dnat_rule" : {
    "id" : "24dd6bf5-48f2-4915-ad0b-5bb111d39c83",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "description" : "aa",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "network_interface_id" : "dae9393a-b536-491c-a5a2-72edc1104707",
    "type" : "COMPUTE",
    "protocol" : "any",
    "internal_service_port" : "0",
    "transit_service_port" : "0",
    "private_ip_address" : "192.168.1.72",
    "created_at" : "2019-04-29T07:10:01",
    "updated_at" : "2019-04-29T07:10:01"
  },
  "request_id" : "70505c941b9b4dfd82fd351932328a2f"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建DNAT规则，其中，中转IP的id为3faa719d-6d18-4ccb-a5c7-33e65a09663e，私网NAT网关实例的id为0adefb29-a6c2-48a5-8637-2be67fa03fec。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreatePrivateDnatSolution {

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



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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePrivateDnatRequest request = new CreatePrivateDnatRequest();
        CreatePrivateDnatOptionBody body = new CreatePrivateDnatOptionBody();
        CreatePrivateDnatOption dnatRulebody = new CreatePrivateDnatOption();
        dnatRulebody.withDescription("aa")
            .withTransitIpId("3faa719d-6d18-4ccb-a5c7-33e65a09663e")
            .withNetworkInterfaceId("dae9393a-b536-491c-a5a2-72edc1104707")
            .withGatewayId("0adefb29-a6c2-48a5-8637-2be67fa03fec");
        body.withDnatRule(dnatRulebody);
        request.withBody(body);
        try {
            CreatePrivateDnatResponse response = client.createPrivateDnat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建DNAT规则，其中，中转IP的id为3faa719d-6d18-4ccb-a5c7-33e65a09663e，私网NAT网关实例的id为0adefb29-a6c2-48a5-8637-2be67fa03fec。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreatePrivateDnatRequest()
        dnatRulebody = CreatePrivateDnatOption(
            description="aa",
```

```
            transit_ip_id="3faa719d-6d18-4ccb-a5c7-33e65a09663e",
            network_interface_id="dae9393a-b536-491c-a5a2-72edc1104707",
            gateway_id="0adefb29-a6c2-48a5-8637-2be67fa03fec"
        )
        request.body = CreatePrivateDnatOptionBody(
            dnat_rule=dnatRulebody
        )
        response = client.create_private_dnat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建DNAT规则，其中，中转IP的id为3faa719d-6d18-4ccb-a5c7-33e65a09663e，私网NAT网关实例的id为0adefb29-a6c2-48a5-8637-2be67fa03fec。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePrivateDnatRequest{}
    descriptionDnatRule:= "aa"
    networkInterfaceIdDnatRule:= "dae9393a-b536-491c-a5a2-72edc1104707"
    dnatRulebody := &model.CreatePrivateDnatOption{
        Description: &descriptionDnatRule,
        TransitIpId: "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
        NetworkInterfaceId: &networkInterfaceIdDnatRule,
        GatewayId: "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    }
    request.Body = &model.CreatePrivateDnatOptionBody{
        DnatRule: dnatRulebody,
    }
    response, err := client.CreatePrivateDnat(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建DNAT规则成功。

错误码

请参见**错误码**。

5.2.4 删除 DNAT 规则

功能介绍

删除指定的DNAT规则。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/private-nat/dnat-rules/{dnat_rule_id}

表 5-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-54 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/dnat-rules/24dd6bf5-48f2-4915-ad0b-5bb111d39c83
```

响应示例

无

状态码

状态码	描述
204	删除DNAT规则成功。

错误码

请参见**错误码**。

5.2.5 查询指定的 DNAT 规则详情

功能介绍

查询指定的DNAT规则详情。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/dnat-rules/{dnat_rule_id}

表 5-55 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
dnat_rule_id	是	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-56 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-57 响应 Body 参数

参数	参数类型	描述
dnat_rule	**PrivateDnat** object	DNAT规则的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-58 PrivateDnat

参数	参数类型	描述
id	String	DNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
description	String	DNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	网络接口ID，支持计算、ELB、VIP等实例的端口。 最小长度：**1** 最大长度：**36**
type	String	DNAT规则后端的类型。 取值： COMPUTE：后端为计算实例。 VIP：后端为VIP的实例。 ELB：后端为ELB的实例。 ELBv3：后端为ELBv3的实例。 CUSTOMIZE：后端为自定义IP。 最小长度：**1** 最大长度：**10**
protocol	String	协议类型。 目前支持TCP/tcp、UDP/udp、ANY/any。 对应协议号6、17、0。 最小长度：**1** 最大长度：**3** 枚举值： • **tcp** • **udp** • **any**

参数	参数类型	描述
private_ip_address	String	后端实例的私网IP地址。 最小长度：**7** 最大长度：**15**
internal_service_port	String	后端实例的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
transit_service_port	String	中转IP的端口号。 最小值：**0** 最大值：**65535** 最小长度：**1** 最大长度：**5**
enterprise_project_id	String	企业项目ID。创建DNAT规则时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**
created_at	String	DNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	DNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**

请求示例

```
GET https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/dnat-rules/
24dd6bf5-48f2-4915-ad0b-5bb111d39c83
```

响应示例

状态码： 200

查询DNAT规则成功。

```
{
  "dnat_rule" : {
    "id" : "24dd6bf5-48f2-4915-ad0b-5bb111d39c83",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "description" : "aa",
    "gateway_id" : "0adefb29-a6c2-48a5-8637-2be67fa03fec",
    "transit_ip_id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
```

```
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "network_interface_id" : "dae9393a-b536-491c-a5a2-72edc1104707",
    "type" : "COMPUTE",
    "protocol" : "any",
    "internal_service_port" : "0",
    "transit_service_port" : "0",
    "private_ip_address" : "192.168.1.72",
    "created_at" : "2019-04-29T07:10:01",
    "updated_at" : "2019-04-29T07:10:01"
  },
  "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```


状态码

状态码	描述
200	查询DNAT规则成功。

错误码

请参见**错误码**。

5.3 SNAT 规则

5.3.1 查询 SNAT 规则列表

功能介绍

查询SNAT规则列表。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/snat-rules

表 5-59 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 5-60 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
marker	否	String	功能说明：分页查询起始的资源ID，为空时查询第一页。 值从上一次查询的PageInfo中的next_marker或者previous_marker中获取。 最小长度：**36** 最大长度：**36**
page_reverse	否	Boolean	是否查询前一页。
id	否	Array	SNAT规则的ID。 数组长度：**1 - 10**
project_id	否	Array	项目的ID。 数组长度：**1 - 10**
description	否	Array	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 数组长度：**1 - 10**
gateway_id	否	Array	私网NAT网关实例的ID。 数组长度：**1 - 10**
cidr	否	Array	规则匹配的CIDR。 数组长度：**1 - 10**
virsubnet_id	否	Array	规则匹配的子网的ID。 数组长度：**1 - 10**
transit_ip_id	否	Array	中转IP的ID。 数组长度：**1 - 10**
transit_ip_address	否	Array	中转IP地址。 数组长度：**1 - 10**
enterprise_project_id	否	Array	企业项目ID。创建SNAT规则时，关联的企业项目ID。 数组长度：**1 - 10**

请求参数

表 5-61 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-62 响应 Body 参数

参数	参数类型	描述
snat_rules	Array of **PrivateSnat** objects	查询SNAT规则列表的响应体。 数组长度：**0 - 2000**
page_info	**PageInfo** object	分页信息。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-63 PrivateSnat

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
gateway_id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	功能说明：规则匹配的CIDR。 取值约束： ● 与virsubnet_id参数二选一。 ● cidr不能与已有snat规则的网段相同。 最小长度：**9** 最大长度：**18**
virsubnet_id	String	功能说明：规则匹配的子网的ID。 取值约束：与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_associations	Array of **AssociatedTransitIp** objects	关联的中转IP详情列表。 数组长度：**1 - 1**
created_at	String	SNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	SNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
enterprise_project_id	String	企业项目id 最小长度：**1** 最大长度：**36**

表 5-64 AssociatedTransitIp

参数	参数类型	描述
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
transit_ip_address	String	中转IP地址。 最小长度：**7** 最大长度：**35**

表 5-65 PageInfo

参数	参数类型	描述
next_marker	String	分页查询结果中最后一条记录的ID。通常用于查询下一页。 最小长度：**1** 最大长度：**36**
previous_marker	String	分页查询结果中第一条记录的ID。通常用于配合page_reverse=true查询上一页。 最小长度：**1** 最大长度：**36**
current_count	Integer	分页查询资源时，本页的实例的个数。 最小值：**1** 最大值：**2000**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat/snat-rules
```

响应示例

状态码： 200

查询SNAT规则列表成功。

```
{
  "snat_rules" : [ {
    "id" : "8a522ff9-8158-494b-83cd-533b045700e6",
    "project_id" : "cfa563efb77d4b6d9960781d82530fd8",
    "description" : "snat rule description",
    "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
    "cidr" : "",
    "virsubnet_id" : "95df1b88-d9bc-4edd-a808-a771dd4ded32",
    "transit_ip_associations" : [ {
      "transit_ip_id" : "bbe7c2e7-3bad-445b-a067-b30acce66053",
      "transit_ip_address" : "172.20.1.98"
    } ],
```

```
  "created_at" : "2019-10-22T03:33:07",
  "updated_at" : "2019-10-22T03:33:07"
 }, {
  "id" : "af4dbb83-7ca0-4ed1-b28b-668c1f9c6b81",
  "project_id" : "cfa563efb77d4b6d9960781d82530fd8",
  "description" : "snat rule description",
  "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
  "cidr" : "",
  "virsubnet_id" : "5b9ea497-727d-4ad0-a99e-3984b3f5aaed",
  "transit_ip_associations" : [ {
    "transit_ip_id" : "36a3049a-1682-48b3-b1cf-cb986a3350ef",
    "transit_ip_address" : "172.20.1.10"
  } ],
  "created_at" : "2019-10-22T03:31:19",
  "updated_at" : "2019-10-22T03:31:19"
 } ],
 "page_info" : {
  "next_marker" : "af4dbb83-7ca0-4ed1-b28b-668c1f9c6b81",
  "previous_marker" : "8a522ff9-8158-494b-83cd-533b045700e6",
  "current_count" : 2
 },
 "request_id" : "69806207-62e3-4950-b463-ff5c1779b714"
}
```


状态码

状态码	描述
200	查询SNAT规则列表成功。

错误码

请参见**错误码**。

5.3.2 查询指定的 SNAT 规则详情

功能介绍

查询指定的SNAT规则详情。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/snat-rules/{snat_rule_id}

表 5-66 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

参数	是否必选	参数类型	描述
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-67 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-68 响应 Body 参数

参数	参数类型	描述
snat_rule	**PrivateSnat** object	SNAT规则的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-69 PrivateSnat

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	功能说明：规则匹配的CIDR。 取值约束： ● 与virsubnet_id参数二选一。 ● cidr不能与已有snat规则的网段相同。 最小长度：**9** 最大长度：**18**
virsubnet_id	String	功能说明：规则匹配的子网的ID。 取值约束：与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_associations	Array of **AssociatedTransitIp** objects	关联的中转IP详情列表。 数组长度：**1 - 1**
created_at	String	SNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	SNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
enterprise_project_id	String	企业项目id 最小长度：**1** 最大长度：**36**

表 5-70 AssociatedTransitIp

参数	参数类型	描述
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
transit_ip_address	String	中转IP地址。 最小长度：**7** 最大长度：**35**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat/snat-rules/8a522ff9-8158-494b-83cd-533b045700e6
```

响应示例

状态码： 200

查询SNAT规则成功。

```
{
  "snat_rule" : {
    "id" : "8a522ff9-8158-494b-83cd-533b045700e6",
    "project_id" : "cfa563efb77d4b6d9960781d82530fd8",
    "description" : "my_snat_rule02",
    "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
    "cidr" : "",
    "virsubnet_id" : "95df1b88-d9bc-4edd-a808-a771dd4ded32",
    "transit_ip_associations" : [ {
      "transit_ip_id" : "bbe7c2e7-3bad-445b-a067-b30acce66053",
      "transit_ip_address" : "172.20.1.98"
    } ],
    "created_at" : "2019-10-22T03:33:07",
    "updated_at" : "2019-10-22T03:33:07"
  },
  "request_id" : "c8b21002-a594-414d-9585-2cc5963d4c3e"
}
```

状态码

状态码	描述
200	查询SNAT规则成功。

错误码

请参见**错误码**。

5.3.3 更新 SNAT 规则

功能介绍

更新指定的SNAT规则。

调用方法

请参见**如何调用API**。

URI

PUT /v3/{project_id}/private-nat/snat-rules/{snat_rule_id}

表 5-71 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-72 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-73 请求 Body 参数

参数	是否必选	参数类型	描述
snat_rule	是	UpdatePrivateSnatOption object	更新SNAT规则的请求体。

表 5-74 UpdatePrivateSnatOption

参数	是否必选	参数类型	描述
transit_ip_ids	否	Array of strings	中转IP的ID的列表。 最小长度：**36** 最大长度：**36** 数组长度：**1 - 1**
description	否	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**

响应参数

状态码： 200

表 5-75 响应 Body 参数

参数	参数类型	描述
request_id	String	请求ID。 最小长度：**36** 最大长度：**36**
snat_rule	PrivateSnat object	SNAT规则的响应体。

表 5-76 PrivateSnat

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

参数	参数类型	描述
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	功能说明：规则匹配的CIDR。 取值约束： ● 与virsubnet_id参数二选一。 ● cidr不能与已有snat规则的网段相同。 最小长度：**9** 最大长度：**18**
virsubnet_id	String	功能说明：规则匹配的子网的ID。 取值约束：与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_associations	Array of **AssociatedTransitIp** objects	关联的中转IP详情列表。 数组长度：**1 - 1**
created_at	String	SNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	SNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
enterprise_project_id	String	企业项目id 最小长度：**1** 最大长度：**36**

表 5-77 AssociatedTransitIp

参数	参数类型	描述
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
transit_ip_address	String	中转IP地址。 最小长度：**7** 最大长度：**35**

请求示例

更新SNAT规则，其中，中转IP的id列表中，有一个id对应的中转IP的id是bbe7c2e7-3bad-445b-a067-b30acce66053，SNAT规则的描述为my_snat_rule_update。

```
PUT https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat/snat-rules/
af4dbb83-7ca0-4ed1-b28b-668c1f9c6b81

{
  "snat_rule" : {
    "description" : "my_snat_rule_update",
    "transit_ip_ids" : [ "bbe7c2e7-3bad-445b-a067-b30acce66053" ]
  }
}
```

响应示例

状态码： 200

更新SNAT规则成功。

```
{
  "request_id" : "15bd32b2-1464-4817-b559-444d22499f6c",
  "snat_rule" : {
    "id" : "af4dbb83-7ca0-4ed1-b28b-668c1f9c6b81",
    "project_id" : "cfa563efb77d4b6d9960781d82530fd8",
    "description" : "my_snat_rule_update",
    "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
    "cidr" : "10.1.1.64/30",
    "virsubnet_id" : "",
    "transit_ip_associations" : [ {
      "transit_ip_id" : "bbe7c2e7-3bad-445b-a067-b30acce66053",
      "transit_ip_address" : "172.20.1.98"
    } ],
    "created_at" : "2019-10-22T03:31:19",
    "updated_at" : "2019-10-22T03:39:52"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新SNAT规则，其中，中转IP的id列表中，有一个id对应的中转IP的id是bbe7c2e7-3bad-445b-a067-b30acce66053，SNAT规则的描述为my_snat_rule_update。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class UpdatePrivateSnatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePrivateSnatRequest request = new UpdatePrivateSnatRequest();
        UpdatePrivateSnatOptionBody body = new UpdatePrivateSnatOptionBody();
        List<String> listSnatRuleTransitIpIds = new ArrayList<>();
        listSnatRuleTransitIpIds.add("bbe7c2e7-3bad-445b-a067-b30acce66053");
        UpdatePrivateSnatOption snatRulebody = new UpdatePrivateSnatOption();
        snatRulebody.withTransitIpIds(listSnatRuleTransitIpIds)
            .withDescription("my_snat_rule_update");
        body.withSnatRule(snatRulebody);
        request.withBody(body);
        try {
            UpdatePrivateSnatResponse response = client.updatePrivateSnat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

更新SNAT规则，其中，中转IP的id列表中，有一个id对应的中转IP的id是bbe7c2e7-3bad-445b-a067-b30acce66053，SNAT规则的描述为my_snat_rule_update。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdatePrivateSnatRequest()
        listTransitIpIdsSnatRule = [
            "bbe7c2e7-3bad-445b-a067-b30acce66053"
        ]
        snatRulebody = UpdatePrivateSnatOption(
            transit_ip_ids=listTransitIpIdsSnatRule,
            description="my_snat_rule_update"
        )
        request.body = UpdatePrivateSnatOptionBody(
            snat_rule=snatRulebody
        )
        response = client.update_private_snat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新SNAT规则，其中，中转IP的id列表中，有一个id对应的中转IP的id是bbe7c2e7-3bad-445b-a067-b30acce66053，SNAT规则的描述为my_snat_rule_update。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePrivateSnatRequest{}
    var listTransitIpIdsSnatRule = []string{
        "bbe7c2e7-3bad-445b-a067-b30acce66053",
    }
    descriptionSnatRule:= "my_snat_rule_update"
    snatRulebody := &model.UpdatePrivateSnatOption{
        TransitIpIds: &listTransitIpIdsSnatRule,
        Description: &descriptionSnatRule,
    }
    request.Body = &model.UpdatePrivateSnatOptionBody{
        SnatRule: snatRulebody,
    }
    response, err := client.UpdatePrivateSnat(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	更新SNAT规则成功。

错误码

请参见**错误码**。

5.3.4 创建 SNAT 规则

功能介绍

创建SNAT规则。

接口约束

创建规则时，要求网关状态status = "ACTIVE"。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat/snat-rules

表 5-78 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-79 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-80 请求 Body 参数

参数	是否必选	参数类型	描述
snat_rule	是	**CreatePrivateSnatOption** object	创建SNAT规则的请求体。

表 5-81 CreatePrivateSnatOption

参数	是否必选	参数类型	描述
gateway_id	是	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	否	String	功能说明：规则匹配的CIDR。 取值约束：与virsubnet_id参数二选一。 最小长度：**9** 最大长度：**18**
virsubnet_id	否	String	功能说明：规则匹配的子网的ID。 取值约束：与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	否	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**0** 最大长度：**255**
transit_ip_ids	是	Array of strings	功能说明：中转IP的ID的列表。 取值约束：中转IP的ID个数不能超过1个。 最小长度：**36** 最大长度：**36** 数组长度：**1 - 1**

响应参数

状态码： 201

表 5-82 响应 Body 参数

参数	参数类型	描述
snat_rule	**PrivateSnat** object	SNAT规则的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-83 PrivateSnat

参数	参数类型	描述
id	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**36** 最大长度：**36**
gateway_id	String	私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
cidr	String	功能说明：规则匹配的CIDR。 取值约束： ● 与virsubnet_id参数二选一。 ● cidr不能与已有snat规则的网段相同。 最小长度：**9** 最大长度：**18**
virsubnet_id	String	功能说明：规则匹配的子网的ID。 取值约束：与cidr参数二选一。 最小长度：**36** 最大长度：**36**
description	String	SNAT规则的描述。长度范围小于等于255个字符，不能包含“<”和“>”。 最小长度：**1** 最大长度：**36**
transit_ip_associations	Array of **AssociatedTransitIp** objects	关联的中转IP详情列表。 数组长度：**1 - 1**
created_at	String	SNAT规则的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**
updated_at	String	SNAT规则的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ。 最小长度：**1** 最大长度：**36**

参数	参数类型	描述
enterprise_project_id	String	企业项目id 最小长度：**1** 最大长度：**36**

表 5-84 AssociatedTransitIp

参数	参数类型	描述
transit_ip_id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
transit_ip_address	String	中转IP地址。 最小长度：**7** 最大长度：**35**

请求示例

创建SNAT规则，其中，SNAT规则的描述为my_snat_rule01，私网NAT网关实例的id为80da6f26-94eb-4537-97f0-5a56f4d04cfb，规则匹配的子网的id为5b9ea497-727d-4ad0-a99e-3984b3f5aaed。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat/snat-rules

{
  "snat_rule" : {
    "description" : "my_snat_rule01",
    "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
    "virsubnet_id" : "5b9ea497-727d-4ad0-a99e-3984b3f5aaed",
    "transit_ip_ids" : [ "36a3049a-1682-48b3-b1cf-cb986a3350ef" ]
  }
}
```

响应示例

状态码： 201

创建SNAT规则成功。

```
{
  "snat_rule" : {
    "id" : "af4dbb83-7ca0-4ed1-b28b-668c1f9c6b81",
    "project_id" : "cfa563efb77d4b6d9960781d82530fd8",
    "description" : "snat rule description",
    "gateway_id" : "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
    "cidr" : "",
    "virsubnet_id" : "5b9ea497-727d-4ad0-a99e-3984b3f5aaed",
    "transit_ip_associations" : [ {
      "transit_ip_id" : "36a3049a-1682-48b3-b1cf-cb986a3350ef",
      "transit_ip_address" : "172.20.1.10"
    } ],
    "created_at" : "2019-10-22T03:31:19",
    "updated_at" : "2019-10-22T03:31:19"
```

```
  },
  "request_id" : "2937502e-73f9-4ba5-ae75-2293a0b35fb8"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建SNAT规则，其中，SNAT规则的描述为my_snat_rule01，私网NAT网关实例的id为80da6f26-94eb-4537-97f0-5a56f4d04cfb，规则匹配的子网的id为5b9ea497-727d-4ad0-a99e-3984b3f5aaed。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class CreatePrivateSnatSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePrivateSnatRequest request = new CreatePrivateSnatRequest();
        CreatePrivateSnatOptionBody body = new CreatePrivateSnatOptionBody();
        List<String> listSnatRuleTransitIpIds = new ArrayList<>();
        listSnatRuleTransitIpIds.add("36a3049a-1682-48b3-b1cf-cb986a3350ef");
        CreatePrivateSnatOption snatRulebody = new CreatePrivateSnatOption();
        snatRulebody.withGatewayId("80da6f26-94eb-4537-97f0-5a56f4d04cfb")
            .withVirsubnetId("5b9ea497-727d-4ad0-a99e-3984b3f5aaed")
            .withDescription("my_snat_rule01")
            .withTransitIpIds(listSnatRuleTransitIpIds);
        body.withSnatRule(snatRulebody);
        request.withBody(body);
        try {
            CreatePrivateSnatResponse response = client.createPrivateSnat(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
```

```
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建SNAT规则，其中，SNAT规则的描述为my_snat_rule01，私网NAT网关实例的id为80da6f26-94eb-4537-97f0-5a56f4d04cfb，规则匹配的子网的id为5b9ea497-727d-4ad0-a99e-3984b3f5aaed。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreatePrivateSnatRequest()
        listTransitIpIdsSnatRule = [
            "36a3049a-1682-48b3-b1cf-cb986a3350ef"
        ]
        snatRulebody = CreatePrivateSnatOption(
            gateway_id="80da6f26-94eb-4537-97f0-5a56f4d04cfb",
            virsubnet_id="5b9ea497-727d-4ad0-a99e-3984b3f5aaed",
            description="my_snat_rule01",
            transit_ip_ids=listTransitIpIdsSnatRule
        )
        request.body = CreatePrivateSnatOptionBody(
            snat_rule=snatRulebody
        )
        response = client.create_private_snat(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建SNAT规则，其中，SNAT规则的描述为my_snat_rule01，私网NAT网关实例的id为80da6f26-94eb-4537-97f0-5a56f4d04cfb，规则匹配的子网的id为5b9ea497-727d-4ad0-a99e-3984b3f5aaed。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePrivateSnatRequest{}
    var listTransitIpIdsSnatRule = []string{
        "36a3049a-1682-48b3-b1cf-cb986a3350ef",
    }
    virsubnetIdSnatRule:= "5b9ea497-727d-4ad0-a99e-3984b3f5aaed"
    descriptionSnatRule:= "my_snat_rule01"
    snatRulebody := &model.CreatePrivateSnatOption{
        GatewayId: "80da6f26-94eb-4537-97f0-5a56f4d04cfb",
        VirsubnetId: &virsubnetIdSnatRule,
        Description: &descriptionSnatRule,
        TransitIpIds: listTransitIpIdsSnatRule,
    }
    request.Body = &model.CreatePrivateSnatOptionBody{
        SnatRule: snatRulebody,
    }
    response, err := client.CreatePrivateSnat(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建SNAT规则成功。

错误码

请参见**错误码**。

5.3.5 删除 SNAT 规则

功能介绍

删除指定的SNAT规则。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/private-nat/snat-rules/{snat_rule_id}

表 5-85 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
snat_rule_id	是	String	SNAT规则的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-86 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat/snat-rules/
8a522ff9-8158-494b-83cd-533b045700e6
```

响应示例

无

状态码

状态码	描述
204	删除SNAT规则成功。

错误码

请参见**错误码**。

5.4 中转 IP

5.4.1 查询中转 IP 列表

功能介绍

查询中转IP列表。

接口约束

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。 支持参数说明中所有非必选参数过滤，请参考请求样例。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/transit-ips

表 5-87 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

表 5-88 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	功能说明：每页返回的个数。 取值范围：0~2000。 默认值：2000。 最小值：**1** 最大值：**2000** 缺省值：**2000**
marker	否	String	功能说明：分页查询起始的资源ID，为空时查询第一页。 值从上一次查询的PageInfo中的next_marker或者previous_marker中获取。 最小长度：**36** 最大长度：**36**
page_reverse	否	Boolean	是否查询前一页。
id	否	Array	中转IP的ID。 数组长度：**1 - 10**
project_id	否	Array	项目的ID。 数组长度：**1 - 10**
network_interface_id	否	Array	中转IP的网络接口ID。 数组长度：**1 - 10**
ip_address	否	Array	中转IP地址。 数组长度：**1 - 10**
gateway_id	否	Array	中转IP绑定的私网NAT网关实例的ID。 数组长度：**1 - 10**
enterprise_project_id	否	Array	企业项目ID。创建中转IP时，关联的企业项目ID。 数组长度：**1 - 10**

参数	是否必选	参数类型	描述
virsubnet_id	否	Array	当前租户子网的ID。 数组长度：**1 - 10**

请求参数

表 5-89 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-90 响应 Body 参数

参数	参数类型	描述
transit_ips	Array of **TransitIp** objects	查询中转IP列表的响应体。 数组长度：**0 - 2000**
page_info	**PageInfo** object	分页信息。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-91 TransitIp

参数	参数类型	描述
id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	中转IP的网络接口ID。 最小长度：**36** 最大长度：**36**
ip_address	String	中转IP的地址。 最小长度：**7** 最大长度：**35**
created_at	String	中转IP的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
updated_at	String	中转IP的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
virsubnet_id	String	当前租户子网的ID。 最小长度：**0** 最大长度：**36**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
gateway_id	String	中转IP绑定的私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。创建中转IP时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-92 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

表 5-93 PageInfo

参数	参数类型	描述
next_marker	String	分页查询结果中最后一条记录的ID。通常用于查询下一页。 最小长度：**1** 最大长度：**36**
previous_marker	String	分页查询结果中第一条记录的ID。通常用于配合page_reverse=true查询上一页。 最小长度：**1** 最大长度：**36**
current_count	Integer	分页查询资源时，本页的实例的个数。 最小值：**1** 最大值：**2000**

请求示例

```
GET https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/transit-ips
```

响应示例

状态码： 200

查询中转IP列表成功。

```
{
  "transit_ips" : [ {
    "id" : "3faa719d-6d18-4ccb-a5c7-33e65a09663e",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "network_interface_id" : "c91c43fb-8d66-48df-bfa9-b89053ac3737",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "gateway_id" : "521bb3d9-8bed-4c6c-9ee8-669bd0620f76",
    "ip_address" : "192.168.1.68",
    "created_at" : "2019-04-29T03:41:59",
    "updated_at" : "2019-04-29T03:41:59",
    "virsubnet_id" : "49ee5fb5-75bf-4320-946e-b21ef4c9c9c1",
    "tags" : [ {
```

```
    "key" : "key1",
    "value" : "value1"
  } ]
 }, {
  "id" : "a2845109-3b2f-4627-b08f-09a726c0a6e7",
  "project_id" : "da261828016849188f4dcc2ef94d9da9",
  "network_interface_id" : "adebbdca-8c26-4c14-b34f-3f53cd2c42f2",
  "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
  "gateway_id" : "521bb3d9-8bed-4c6c-9ee8-669bd0620f76",
  "ip_address" : "192.168.1.68",
  "created_at" : "2019-04-29T02:16:09",
  "updated_at" : "2019-04-29T02:16:09",
  "virsubnet_id" : "333e5fb5-75bf-4320-946e-b21ef4c9c2g5",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  } ]
 } ],
 "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```


SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListTransitIpsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListTransitIpsRequest request = new ListTransitIpsRequest();
        request.withLimit(<limit>);
        request.withMarker("<marker>");
        request.withPageReverse(<page_reverse>);
        request.withId();
        request.withNetworkInterfaceId();
```



```
        request.withIpAddress();
        request.withGatewayId();
        request.withEnterpriseProjectId();
        request.withVirsubnetId();
        try {
            ListTransitIpsResponse response = client.listTransitIps(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ListTransitIpsRequest()
        request.limit = <limit>
        request.marker = "<marker>"
        request.page_reverse = <PageReverse>
        request.id =
        request.network_interface_id =
        request.ip_address =
        request.gateway_id =
        request.enterprise_project_id =
        request.virsubnet_id =
        response = client.list_transit_ips(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTransitIpsRequest{}
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    markerRequest:= "<marker>"
    request.Marker = &markerRequest
    pageReverseRequest:= <page_reverse>
    request.PageReverse = &pageReverseRequest
    response, err := client.ListTransitIps(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询中转IP列表成功。

错误码

请参见**错误码**。

5.4.2 删除中转 IP

功能介绍

删除中转IP。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/private-nat/transit-ips/{transit_ip_id}

表 5-94 路径参数

参数	是否必选	参数类型	描述
transit_ip_id	是	String	中转IP的ID。 最小长度：**1** 最大长度：**36**
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-95 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**512**

响应参数

无

请求示例

```
DELETE https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/transit-ips/
a2845109-3b2f-4627-b08f-09a726c0a6e7
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeleteTransitIpSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteTransitIpRequest request = new DeleteTransitIpRequest();
        try {
            DeleteTransitIpResponse response = client.deleteTransitIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteTransitIpRequest{}
    response, err := client.DeleteTransitIp(request)
```

```
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除中转IP成功。

错误码

请参见**错误码**。

5.4.3 创建中转 IP

功能介绍

创建中转IP。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat/transit-ips

表 5-96 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**

请求参数

表 5-97 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**512**

表 5-98 请求 Body 参数

参数	是否必选	参数类型	描述
transit_ip	是	**CreatTransitIpOption** object	创建中转IP的请求体。

表 5-99 CreatTransitIpOption

参数	是否必选	参数类型	描述
virsubnet_id	是	String	当前项目子网的ID。 最小长度：**1** 最大长度：**36**
ip_address	否	String	中转IP地址。 最小长度：**7** 最大长度：**35**
enterprise_project_id	否	String	企业项目ID。创建中转IP时，关联的企业项目ID。 缺省值：**0** 最小长度：**1** 最大长度：**36**
tags	否	Array of **Tag** objects	标签 数组长度：**0 - 2000**

表 5-100 Tag

参数	是否必选	参数类型	描述
key	是	String	标签key值。 最小长度：**1** 最大长度：**128**
value	是	String	标签value。 最小长度：**0** 最大长度：**255**

响应参数

状态码： 201

表 5-101 响应 Body 参数

参数	参数类型	描述
transit_ip	**TransitIp** object	中转子网IP的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-102 TransitIp

参数	参数类型	描述
id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	中转IP的网络接口ID。 最小长度：**36** 最大长度：**36**
ip_address	String	中转IP的地址。 最小长度：**7** 最大长度：**35**

参数	参数类型	描述
created_at	String	中转IP的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
updated_at	String	中转IP的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
virsubnet_id	String	当前租户子网的ID。 最小长度：**0** 最大长度：**36**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
gateway_id	String	中转IP绑定的私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。创建中转IP时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-103 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

创建中转IP，其中，当前项目子网的id为2759da7b-8015-404c-ae0a-a389007b0e2a，中转IP地址为192.168.1.68，创建中转IP时，关联的企业项目id为2759da7b-8015-404c-ae0a-a389007b0e2a。

```
POST https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/transit-ips
```

```
{
  "transit_ip" : {
    "virsubnet_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "ip_address" : "192.168.1.68",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ]
  }
}
```

响应示例

状态码： 201

创建中转IP成功。

```
{
  "transit_ip" : {
    "id" : "a2845109-3b2f-4627-b08f-09a726c0a6e7",
    "project_id" : "da2618280168491 88f4dcc2ef94d9da9",
    "network_interface_id" : "adebbdca-8c26-4c14-b34f-3f53cd2c42f2",
    "ip_address" : "192.168.1.68",
    "gateway_id" : "521bb3d9-8bed-4c6c-9ee8-669bd0620f76",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "created_at" : "2019-04-29T02:16:09",
    "updated_at" : "2019-04-29T02:16:09",
    "virsubnet_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ]
  },
  "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建中转IP，其中，当前项目子网的id为2759da7b-8015-404c-ae0a-a389007b0e2a，中转IP地址为192.168.1.68，创建中转IP时，关联的企业项目id为2759da7b-8015-404c-ae0a-a389007b0e2a。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class CreateTransitIpSolution {

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



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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateTransitIpRequest request = new CreateTransitIpRequest();
        CreateTransitIpRequestBody body = new CreateTransitIpRequestBody();
        List<PrivateTag> listTransitIpTags = new ArrayList<>();
        listTransitIpTags.add(
            new PrivateTag()
                .withKey("key1")
                .withValue("value1")
        );
        CreatTransitIpOption transitIpbody = new CreatTransitIpOption();
        transitIpbody.withVirsubnetId("2759da7b-8015-404c-ae0a-a389007b0e2a")
            .withIpAddress("192.168.1.68")
            .withEnterpriseProjectId("2759da7b-8015-404c-ae0a-a389007b0e2a")
            .withTags(listTransitIpTags);
        body.withTransitIp(transitIpbody);
        request.withBody(body);
        try {
            CreateTransitIpResponse response = client.createTransitIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

创建中转IP，其中，当前项目子网的id为2759da7b-8015-404c-ae0a-a389007b0e2a，中转IP地址为192.168.1.68，创建中转IP时，关联的企业项目id为2759da7b-8015-404c-ae0a-a389007b0e2a。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

```
    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateTransitIpRequest()
        listTagsTransitIp = [
            PrivateTag(
                key="key1",
                value="value1"
            )
        ]
        transitIpbody = CreatTransitIpOption(
            virsubnet_id="2759da7b-8015-404c-ae0a-a389007b0e2a",
            ip_address="192.168.1.68",
            enterprise_project_id="2759da7b-8015-404c-ae0a-a389007b0e2a",
            tags=listTagsTransitIp
        )
        request.body = CreateTransitIpRequestBody(
            transit_ip=transitIpbody
        )
        response = client.create_transit_ip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建中转IP，其中，当前项目子网的id为2759da7b-8015-404c-ae0a-a389007b0e2a，中转IP地址为192.168.1.68，创建中转IP时，关联的企业项目id为2759da7b-8015-404c-ae0a-a389007b0e2a。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.CreateTransitIpRequest{}
    var listTagsTransitIp = []model.PrivateTag{
        {
            Key: "key1",
            Value: "value1",
        },
    }
    ipAddressTransitIp:= "192.168.1.68"
    enterpriseProjectIdTransitIp:= "2759da7b-8015-404c-ae0a-a389007b0e2a"
    transitIpbody := &model.CreatTransitIpOption{
        VirsubnetId: "2759da7b-8015-404c-ae0a-a389007b0e2a",
        IpAddress: &ipAddressTransitIp,
        EnterpriseProjectId: &enterpriseProjectIdTransitIp,
        Tags: &listTagsTransitIp,
    }
    request.Body = &model.CreateTransitIpRequestBody{
        TransitIp: transitIpbody,
    }
    response, err := client.CreateTransitIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	创建中转IP成功。

错误码

请参见**错误码**。

5.4.4 查询指定的中转 IP 详情

功能介绍

查询中转IP详情。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat/transit-ips/{transit_ip_id}

表 5-104 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**36**
transit_ip_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-105 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**512**

响应参数

状态码： 200

表 5-106 响应 Body 参数

参数	参数类型	描述
transit_ip	**TransitIp** object	中转子网IP的响应体。
request_id	String	请求ID。 最小长度：**1** 最大长度：**36**

表 5-107 TransitIp

参数	参数类型	描述
id	String	中转IP的ID。 最小长度：**36** 最大长度：**36**
project_id	String	项目的ID。 最小长度：**1** 最大长度：**36**
network_interface_id	String	中转IP的网络接口ID。 最小长度：**36** 最大长度：**36**
ip_address	String	中转IP的地址。 最小长度：**7** 最大长度：**35**
created_at	String	中转IP的创建时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
updated_at	String	中转IP的更新时间，遵循UTC时间，格式是yyyy-mm-ddThh:mm:ssZ 最小长度：**1** 最大长度：**36**
virsubnet_id	String	当前租户子网的ID。 最小长度：**0** 最大长度：**36**
tags	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**
gateway_id	String	中转IP绑定的私网NAT网关实例的ID。 最小长度：**36** 最大长度：**36**
enterprise_project_id	String	企业项目ID。创建中转IP时，关联的企业项目ID。 最小长度：**1** 最大长度：**36**

表 5-108 Tag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/da261828016849188f4dcc2ef94d9da9/private-nat/transit-ips/
a2845109-3b2f-4627-b08f-09a726c0a6e7
```

响应示例

状态码： 200

查询指定中转IP成功。

```
{
  "transit_ip" : {
    "id" : "a2845109-3b2f-4627-b08f-09a726c0a6e7",
    "project_id" : "da261828016849188f4dcc2ef94d9da9",
    "network_interface_id" : "adebbdca-8c26-4c14-b34f-3f53cd2c42f2",
    "ip_address" : "192.168.1.68",
    "gateway_id" : "521bb3d9-8bed-4c6c-9ee8-669bd0620f76",
    "enterprise_project_id" : "2759da7b-8015-404c-ae0a-a389007b0e2a",
    "created_at" : "2019-04-29T02:16:09",
    "updated_at" : "2019-04-29T02:16:09",
    "virsubnet_id" : "49ee5fb5-75bf-4320-946e-b21ef4c9c9c1",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    } ]
  },
  "request_id" : "747a911c17067a39692f75ac146fb47e"
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;
```

```
public class ShowTransitIpSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTransitIpRequest request = new ShowTransitIpRequest();
        try {
            ShowTransitIpResponse response = client.showTransitIp(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowTransitIpRequest()
        response = client.show_transit_ip(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询指定中转IP成功。

错误码

请参见**错误码**。

5.5 私网 NAT 网关标签管理

5.5.1 查询私网 NAT 网关实例

功能介绍

- 使用标签过滤私网NAT网关实例。
- 标签管理服务需要提供按标签过滤私网NAT网关服务实例并汇总显示在列表中，需要私网NAT网关服务提供查询能力。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat-gateways/resource_instances/action

表 5-109 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**

请求参数

表 5-110 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-111 请求 Body 参数

参数	是否必选	参数类型	描述
offset	否	String	索引位置， 从offset指定的下一条数据开始查询。 查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据时响应体中的值带入此参数（action为count时无此参数）如果action为filter默认为0，必须为数字，不能为负数。 最小长度：**0** 最大长度：**65535**
limit	否	String	查询记录数（action为count时无此参数）如果action为filter默认为1000，limit最多为1000,不能为负数，最小值为1。 最小长度：**1** 最大长度：**1000**
action	是	String	操作标识（仅限于filter，count）： filter（过滤），count(查询总条数) 如果是filter就是分页查询，如果是count只需按照条件将总条数返回即可。 枚举值： • **filter** • **count**
matches	否	Array of **Match** objects	搜索字段,key为要匹配的字段，如resource_name等。value为匹配的值。此字段为固定字典值。 根据不同的字段确认是否需要模糊匹配，如resource_name默认为模糊搜索（不区分大小写），如果value为空字符串精确匹配。resource_id为精确匹配。第一期只做resource_name，后续在扩展。

参数	是否必选	参数类型	描述
not_tags	否	Array of **Tags** objects	不包含标签，最多包含10个key，每个key下面的value最多10个, 结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**
tags	否	Array of **Tags** objects	包含标签，最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：**1 - 10**
tags_any	否	Array of **Tags** objects	包含任意标签，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含任意标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**
not_tags_any	否	Array of **Tags** objects	不包含任意标签，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含任意标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**

表 5-112 Match

参数	是否必选	参数类型	描述
key	是	String	搜索条件key值。 最小长度：**1** 最大长度：**128**
value	是	String	搜索条件value。 最小长度：**0** 最大长度：**255**

表 5-113 Tags

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。搜索时不对此参数做校验，key不能为空或者空字符串，不能为空格，校验和使用之前先 trim 前后空格。 最小长度：**1** 最大长度：**128**
values	是	Array of strings	值列表。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

状态码： 200

表 5-114 响应 Body 参数

参数	参数类型	描述
resources	Array of **Resource** objects	资源列表。 数组长度：**1 - 2000**
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
total_count	Integer	总记录数。

表 5-115 Resource

参数	参数类型	描述
resource_detail	Object	资源详情。用于扩展。默认为空。
resource_id	String	资源的ID。 最小长度：**36** 最大长度：**36**
resource_name	String	资源名称，资源没有名称时默认为空字符串。 最小长度：**0** 最大长度：**36**
resource_tag	Array of **ResourceTag** objects	标签列表，没有标签默认为空数组。 数组长度：**1 - 10**

表 5-116 ResourceTag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

- 查询私网NAT网关实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
POST  https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/resource_instances/action

{
  "offset" : "10",
  "limit" : "10",
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
```

```
 "tags_any" : [ {
   "key" : "key1",
   "values" : [ "value1", "value2" ]
 } ],
 "not_tags_any" : [ {
   "key" : "key1",
   "values" : [ "value1", "value2" ]
 } ]
}
```

- 查询私网NAT网关实例，其中，操作标识为count，按照条件将总条数返回。

```
POST  https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/
resource_instances/action

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

响应示例

状态码： 200

- 查询操作成功。
- 示例1：action为count时的响应体
- 示例2：action为filter时的响应体
- 示例 1

```
{
 "request_id" : "a67262f6b7242d63d4ae95e41abf2790",
 "total_count" : 100
}
```

- 示例 2

```
{
 "resources" : [ {
   "resource_detail" : null,
   "resource_id" : "e5ad289f-9c56-4daf-b08b-2e53a983473a",
   "resource_name" : "nat_gateways",
   "tags" : [ {
     "key" : "key1",
     "value" : "value1"
   }, {
     "key" : "key2",
```

```
    "value" : "value1"
  } ]
 } ],
 "request_id" : "a67262f6b7242d63d4ae95e41abf2790",
 "total_count" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 查询私网NAT网关实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListPrivateNatsByTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPrivateNatsByTagsRequest request = new ListPrivateNatsByTagsRequest();
        ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<Tags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<Tags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new Tags()
                .withKey("key1")
```

```
            .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("*value1");
        listTagsValues.add("value2");
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("*value1");
        listNotTagsValues.add("value2");
        List<Tags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<Match> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new Match()
                .withKey("resource_name")
                .withValue("resource1")
        );
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("filter"));
        body.withLimit("10");
        body.withOffset("10");
        request.withBody(body);
        try {
            ListPrivateNatsByTagsResponse response = client.listPrivateNatsByTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 查询私网NAT网关实例，其中，操作标识为count，按照条件将总条数返回。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListPrivateNatsByTagsSolution {

    public static void main(String[] args) {
```



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

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

    NatClient client = NatClient.newBuilder()
            .withCredential(auth)
            .withRegion(NatRegion.valueOf("<YOUR REGION>"))
            .build();
    ListPrivateNatsByTagsRequest request = new ListPrivateNatsByTagsRequest();
    ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
    List<String> listNotTagsAnyValues = new ArrayList<>();
    listNotTagsAnyValues.add("value1");
    listNotTagsAnyValues.add("value2");
    List<Tags> listbodyNotTagsAny = new ArrayList<>();
    listbodyNotTagsAny.add(
        new Tags()
            .withKey("key1")
            .withValues(listNotTagsAnyValues)
    );
    List<String> listTagsAnyValues = new ArrayList<>();
    listTagsAnyValues.add("value1");
    listTagsAnyValues.add("value2");
    List<Tags> listbodyTagsAny = new ArrayList<>();
    listbodyTagsAny.add(
        new Tags()
            .withKey("key1")
            .withValues(listTagsAnyValues)
    );
    List<String> listTagsValues = new ArrayList<>();
    listTagsValues.add("value1");
    listTagsValues.add("value2");
    List<String> listTagsValues1 = new ArrayList<>();
    listTagsValues1.add("value1");
    listTagsValues1.add("value2");
    List<Tags> listbodyTags = new ArrayList<>();
    listbodyTags.add(
        new Tags()
            .withKey("key1")
            .withValues(listTagsValues1)
    );
    listbodyTags.add(
        new Tags()
            .withKey("key2")
            .withValues(listTagsValues)
    );
    List<String> listNotTagsValues = new ArrayList<>();
    listNotTagsValues.add("value1");
    listNotTagsValues.add("*value2");
    List<Tags> listbodyNotTags = new ArrayList<>();
    listbodyNotTags.add(
        new Tags()
            .withKey("key1")
            .withValues(listNotTagsValues)
    );
    List<Match> listbodyMatches = new ArrayList<>();
    listbodyMatches.add(
        new Match()
            .withKey("resource_name")
            .withValue("resource1")
    );
```

```
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("count"));
        request.withBody(body);
        try {
            ListPrivateNatsByTagsResponse response = client.listPrivateNatsByTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 查询私网NAT网关实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListPrivateNatsByTagsRequest()
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
```

```
        listTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "*value1",
            "value2"
        ]
        listTagsbody = [
            Tags(
                key="key1",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "*value1",
            "value2"
        ]
        listNotTagsbody = [
            Tags(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            Match(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ListTagResourceInstancesRequestBody(
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody,
            action="filter",
            limit="10",
            offset="10"
        )
        response = client.list_private_nats_by_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 查询私网NAT网关实例，其中，操作标识为count，按照条件将总条数返回。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \
```

```
client = NatClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(NatRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListPrivateNatsByTagsRequest()
    listValuesNotTagsAny = [
        "value1",
        "value2"
    ]
    listNotTagsAnybody = [
        Tags(
            key="key1",
            values=listValuesNotTagsAny
        )
    ]
    listValuesTagsAny = [
        "value1",
        "value2"
    ]
    listTagsAnybody = [
        Tags(
            key="key1",
            values=listValuesTagsAny
        )
    ]
    listValuesTags = [
        "value1",
        "value2"
    ]
    listValuesTags1 = [
        "value1",
        "value2"
    ]
    listTagsbody = [
        Tags(
            key="key1",
            values=listValuesTags1
        ),
        Tags(
            key="key2",
            values=listValuesTags
        )
    ]
    listValuesNotTags = [
        "value1",
        "*value2"
    ]
    listNotTagsbody = [
        Tags(
            key="key1",
            values=listValuesNotTags
        )
    ]
    listMatchesbody = [
        Match(
            key="resource_name",
            value="resource1"
        )
    ]
    request.body = ListTagResourceInstancesRequestBody(
        not_tags_any=listNotTagsAnybody,
        tags_any=listTagsAnybody,
        tags=listTagsbody,
        not_tags=listNotTagsbody,
        matches=listMatchesbody,
        action="count"
    )
```

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

Go

- 查询私网NAT网关实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPrivateNatsByTagsRequest{}
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
```

```
    var listTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listMatchesbody = []model.Match{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    limitListTagResourceInstancesRequestBody:= "10"
    offsetListTagResourceInstancesRequestBody:= "10"
    request.Body = &model.ListTagResourceInstancesRequestBody{
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
        Action: model.GetListTagResourceInstancesRequestBodyActionEnum().FILTER,
        Limit: &limitListTagResourceInstancesRequestBody,
        Offset: &offsetListTagResourceInstancesRequestBody,
    }
    response, err := client.ListPrivateNatsByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 查询私网NAT网关实例，其中，操作标识为count，按照条件将总条数返回。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListPrivateNatsByTagsRequest{}
var listValuesNotTagsAny = []string{
    "value1",
    "value2",
}
var listNotTagsAnybody = []model.Tags{
    {
        Key: "key1",
        Values: listValuesNotTagsAny,
    },
}
var listValuesTagsAny = []string{
    "value1",
    "value2",
}
var listTagsAnybody = []model.Tags{
    {
        Key: "key1",
        Values: listValuesTagsAny,
    },
}
var listValuesTags = []string{
    "value1",
    "value2",
}
var listValuesTags1 = []string{
    "value1",
    "value2",
}
var listTagsbody = []model.Tags{
    {
        Key: "key1",
        Values: listValuesTags1,
    },
    {
        Key: "key2",
        Values: listValuesTags,
    },
}
var listValuesNotTags = []string{
    "value1",
    "*value2",
}
var listNotTagsbody = []model.Tags{
    {
        Key: "key1",
        Values: listValuesNotTags,
    },
}
var listMatchesbody = []model.Match{
    {
        Key: "resource_name",
        Value: "resource1",
    },
}
request.Body = &model.ListTagResourceInstancesRequestBody{
    NotTagsAny: &listNotTagsAnybody,
    TagsAny: &listTagsAnybody,
    Tags: &listTagsbody,
    NotTags: &listNotTagsbody,
    Matches: &listMatchesbody,
    Action: model.GetListTagResourceInstancesRequestBodyActionEnum().COUNT,
}
response, err := client.ListPrivateNatsByTags(request)
if err == nil {
```

```
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	• 查询操作成功。 • 示例1：action为count时的响应体 • 示例2：action为filter时的响应体

错误码

请参见**错误码**。

5.5.2 查询私网 NAT 网关项目标签

功能介绍

- 查询租户在指定Project的所有私网NAT网关标签集合。
- 标签管理服务需要能够列出当前租户全部已使用的私网NAT网关标签集合，为打私网NAT网关标签和过滤私网NAT网关实例时提供标签联想功能。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat-gateways/tags

表 5-117 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**

请求参数

表 5-118 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-119 响应 Body 参数

参数	参数类型	描述
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
tags	Array of **Tags** objects	标签。 数组长度：**1 - 10**

表 5-120 Tags

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。搜索时不对此参数做校验，key不能为空或者空字符串，不能为空格，校验和使用之前先trim 前后空格。 最小长度：**1** 最大长度：**128**
values	Array of strings	值列表。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/tags
```

响应示例

状态码： 200

查询操作成功。

```
{
  "request_id" : "c285190c-b9e9-4f38-a69a-6745f22d8dca",
  "tags" : [ {
    "key" : "keys",
    "values" : [ "value" ]
  }, {
    "key" : "key3",
    "values" : [ "value3", "value33" ]
  }, {
    "key" : "key1",
    "values" : [ "value1" ]
  }, {
    "key" : "key2",
    "values" : [ "value2", "value22" ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ListPrivateNatTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPrivateNatTagsRequest request = new ListPrivateNatTagsRequest();
        try {
            ListPrivateNatTagsResponse response = client.listPrivateNatTags(request);
```

```
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

```
    sk := os.Getenv("CLOUD_SDK_SK")

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

5.5.3 查询私网 NAT 网关标签

功能介绍

- 查询指定私网NAT网关实例的标签信息。
- 标签管理服务需要使用该接口查询指定私网NAT网关实例的全部标签数据。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/private-nat-gateways/{resource_id}/tags

表 5-121 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	私网NAT网关的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-122 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-123 响应 Body 参数

参数	参数类型	描述
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
tags	Array of **Tag** objects	标签。 数组长度：**1 - 10**

表 5-124 Tag

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/b0399473-c352-4d0c-8ff4-cfde719cfde9/tags
```

响应示例

状态码： 200

查询操作成功。

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

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ShowPrivateNatTagsSolution {

    public static void main(String[] args) {
```

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPrivateNatTagsRequest request = new ShowPrivateNatTagsRequest();
        try {
            ShowPrivateNatTagsResponse response = client.showPrivateNatTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

5.5.4 添加私网 NAT 网关标签

功能介绍

- 一个私网NAT网关上最多有10个标签。
- 此接口为幂等接口：
- 创建时，如果创建的标签已经存在（key相同），则覆盖。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat-gateways/{resource_id}/tags

表 5-125 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	私网NAT网关的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-126 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-127 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	**Tag** object	标签。

表 5-128 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	是	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

无

请求示例

添加私网NAT网关标签，其中，标签键为“key1”，对应的值为“value1”。

```
POST  https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags

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

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

添加私网NAT网关标签，其中，标签键为“key1”，对应的值为“value1”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreatePrivateNatTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePrivateNatTagRequest request = new CreatePrivateNatTagRequest();
        CreateResourceTagRequestBody body = new CreateResourceTagRequestBody();
        Tag tagbody = new Tag();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreatePrivateNatTagResponse response = client.createPrivateNatTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

添加私网NAT网关标签，其中，标签键为“key1”，对应的值为“value1”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreatePrivateNatTagRequest()
        tagbody = Tag(
            key="key1",
            value="value1"
        )
        request.body = CreateResourceTagRequestBody(
            tag=tagbody
        )
        response = client.create_private_nat_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

添加私网NAT网关标签，其中，标签键为“key1”，对应的值为“value1”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePrivateNatTagRequest{}
    tagbody := &model.Tag{
        Key: "key1",
        Value: "value1",
    }
    request.Body = &model.CreateResourceTagRequestBody{
        Tag: tagbody,
    }
```

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	添加操作成功。

错误码

请参见**错误码**。

5.5.5 批量添加删除私网 NAT 网关标签

功能介绍

- 为指定私网NAT网关实例批量添加或删除标签
- 标签管理服务需要使用该接口批量管理私网NAT网关实例的标签。
- 一个私网NAT网关上最多有10个标签。

接口约束

此接口为幂等接口：

- 创建时如果请求体中存在重复key则报错。
- 创建时，不允许设置重复key数据,如果数据库已存在该key，就覆盖value的值。
- 删除时，如果删除的标签不存在，默认处理成功，删除时不对标签字符集范围做校验。
- 删除时tags结构体不能缺失，key不能为空，或者空字符串。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/private-nat-gateways/{resource_id}/tags/action

表 5-129 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	私网NAT网关的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-130 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-131 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	功能说明：操作标识。 取值范围： create（创建） delete（删除） 枚举值： • **create** • **delete**
tags	是	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**

表 5-132 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	是	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

无

请求示例

- 批量添加私网NAT网关标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

- 批量删除私网NAT网关标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/action

{
  "action" : "delete",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

- 批量添加私网NAT网关标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeletePrivateNatTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateDeletePrivateNatTagsRequest request = new BatchCreateDeletePrivateNatTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateDeletePrivateNatTagsResponse response = client.batchCreateDeletePrivateNatTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```



```
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 批量删除私网NAT网关标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeletePrivateNatTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateDeletePrivateNatTagsRequest request = new BatchCreateDeletePrivateNatTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchCreateDeletePrivateNatTagsResponse response = client.batchCreateDeletePrivateNatTags(request);
            System.out.println(response.toString());
```



```
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 批量添加私网NAT网关标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeletePrivateNatTagsRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_delete_private_nat_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 批量删除私网NAT网关标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeletePrivateNatTagsRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="delete"
        )
        response = client.batch_create_delete_private_nat_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 批量添加私网NAT网关标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

func main() {
```

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeletePrivateNatTagsRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().CREATE,
    }
    response, err := client.BatchCreateDeletePrivateNatTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 批量删除私网NAT网关标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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



```
        Build()

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeletePrivateNatTagsRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().DELETE,
    }
    response, err := client.BatchCreateDeletePrivateNatTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	批量添加或删除操作成功。

错误码

请参见**错误码**。

5.5.6 删除私网 NAT 网关标签

功能介绍

- 幂等接口：
- 删除时，不对标签字符集做校验，调用接口前必须要做encodeURI，服务端需要对接口uri做decodeURI。删除的key不存在报404，key不能为空或者空字符串。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/private-nat-gateways/{resource_id}/tags/{key}

表 5-133 路径参数

参数	是否必选	参数类型	描述
key	是	String	标签key。 最小长度：**1** 最大长度：**128**
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	私网NAT网关的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-134 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

删除私网NAT网关标签。

```
DELETE https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/private-nat-gateways/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/key1
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeletePrivateNatTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePrivateNatTagRequest request = new DeletePrivateNatTagRequest();
        try {
            DeletePrivateNatTagResponse response = client.deletePrivateNatTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *
```

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除操作成功。

错误码

请参见**错误码**。

5.6 中转 IP 标签管理

5.6.1 查询中转 IP 实例

功能介绍

- 使用标签过滤中转IP实例。
- 标签管理服务需要提供按标签过滤中转IP服务实例并汇总显示在列表中，需要中转IP服务提供查询能力。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/transit-ips/resource_instances/action

表 5-135 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**

请求参数

表 5-136 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-137 请求 Body 参数

参数	是否必选	参数类型	描述
offset	否	String	索引位置， 从offset指定的下一条数据开始查询。 查询第一页数据时，不需要传入此参数，查询后续页码数据时，将查询前一页数据时响应体中的值带入此参数（action为count时无此参数）如果action为filter默认为0，必须为数字，不能为负数。 最小长度：**0** 最大长度：**65535**
limit	否	String	查询记录数（action为count时无此参数）如果action为filter默认为1000，limit最多为1000,不能为负数，最小值为1。 最小长度：**1** 最大长度：**1000**
action	是	String	操作标识（仅限于filter，count）： filter（过滤），count(查询总条数) 如果是filter就是分页查询，如果是count只需按照条件将总条数返回即可。 枚举值： • **filter** • **count**

参数	是否必选	参数类型	描述
matches	否	Array of **Match** objects	搜索字段,key为要匹配的字段，如resource_name等。value为匹配的值。此字段为固定字典值。 根据不同的字段确认是否需要模糊匹配，如resource_name默认为模糊搜索（不区分大小写），如果value为空字符串精确匹配。resource_id为精确匹配。第一期只做resource_name，后续在扩展。
not_tags	否	Array of **Tags** objects	不包含标签，最多包含10个key，每个key下面的value最多10个, 结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**
tags	否	Array of **Tags** objects	包含标签，最多包含10个key，每个key下面的value最多10个，结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含所有标签的资源列表，key之间是与的关系，key-value结构中value是或的关系。无tag过滤条件时返回全量数据。 数组长度：**1 - 10**
tags_any	否	Array of **Tags** objects	包含任意标签，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回包含任意标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**

参数	是否必选	参数类型	描述
not_tags_any	否	Array of **Tags** objects	不包含任意标签，最多包含10个key，每个key下面的value最多10个,结构体不能缺失，key不能为空或者空字符串。Key不能重复，同一个key中values不能重复。返回不包含任意标签的资源列表，key之间是或的关系，key-value结构中value是或的关系。无过滤条件时返回全量数据。 数组长度：**1 - 10**

表 5-138 Match

参数	是否必选	参数类型	描述
key	是	String	搜索条件key值。 最小长度：**1** 最大长度：**128**
value	是	String	搜索条件value。 最小长度：**0** 最大长度：**255**

表 5-139 Tags

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。搜索时不对此参数做校验，key不能为空或者空字符串，不能为空格，校验和使用之前先trim 前后空格。 最小长度：**1** 最大长度：**128**
values	是	Array of strings	值列表。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

状态码： 200

表 5-140 响应 Body 参数

参数	参数类型	描述
resources	Array of **Resource** objects	资源列表。 数组长度：**1 - 2000**
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
total_count	Integer	总记录数。

表 5-141 Resource

参数	参数类型	描述
resource_detail	Object	资源详情。用于扩展。默认为空。
resource_id	String	资源的ID。 最小长度：**36** 最大长度：**36**
resource_name	String	资源名称，资源没有名称时默认为空字符串。 最小长度：**0** 最大长度：**36**
resource_tag	Array of **ResourceTag** objects	标签列表，没有标签默认为空数组。 数组长度：**1 - 10**

表 5-142 ResourceTag

参数	参数类型	描述
key	String	标签key值。 最小长度：**1** 最大长度：**128**
value	String	标签value。 最小长度：**0** 最大长度：**255**

请求示例

- 查询中转IP实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/resource_instances/action

{
  "offset" : "10",
  "limit" : "10",
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "resource1"
  } ],
  "not_tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags" : [ {
    "key" : "key1",
    "values" : [ "*value1", "value2" ]
  } ],
  "tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ],
  "not_tags_any" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  } ]
}
```

- 查询中转IP实例，其中，操作标识为count，按照条件将总条数返回。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/resource_instances/action

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

响应示例

状态码： 200

- 查询操作成功。
- 示例1：action为count时的响应体
- 示例2：action为filter时的响应体
- 示例 1

```
{
  "request_id" : "d70aabc854d3d301f9bb106e6b70ac99",
  "total_count" : 100
}
```

- 示例 2

```
{
  "resources" : [ {
    "resource_detail" : null,
    "resource_id" : "ae33be9b-d2c0-441b-a8d0-f6dafedf1778",
    "resource_name" : "transit_ips",
    "tags" : [ {
      "key" : "key1",
      "value" : "value1"
    }, {
      "key" : "key2",
      "value" : "value1"
    } ]
  } ],
  "request_id" : "9e47d9476cfd346f864cb77acb274185",
  "total_count" : 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 查询中转IP实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListTransitIpsByTagsSolution {

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

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

```
NatClient client = NatClient.newBuilder()
        .withCredential(auth)
        .withRegion(NatRegion.valueOf("<YOUR REGION>"))
        .build();
ListTransitIpsByTagsRequest request = new ListTransitIpsByTagsRequest();
ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
List<String> listNotTagsAnyValues = new ArrayList<>();
listNotTagsAnyValues.add("value1");
listNotTagsAnyValues.add("value2");
List<Tags> listbodyNotTagsAny = new ArrayList<>();
listbodyNotTagsAny.add(
    new Tags()
        .withKey("key1")
        .withValues(listNotTagsAnyValues)
);
List<String> listTagsAnyValues = new ArrayList<>();
listTagsAnyValues.add("value1");
listTagsAnyValues.add("value2");
List<Tags> listbodyTagsAny = new ArrayList<>();
listbodyTagsAny.add(
    new Tags()
        .withKey("key1")
        .withValues(listTagsAnyValues)
);
List<String> listTagsValues = new ArrayList<>();
listTagsValues.add("*value1");
listTagsValues.add("value2");
List<Tags> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tags()
        .withKey("key1")
        .withValues(listTagsValues)
);
List<String> listNotTagsValues = new ArrayList<>();
listNotTagsValues.add("*value1");
listNotTagsValues.add("value2");
List<Tags> listbodyNotTags = new ArrayList<>();
listbodyNotTags.add(
    new Tags()
        .withKey("key1")
        .withValues(listNotTagsValues)
);
List<Match> listbodyMatches = new ArrayList<>();
listbodyMatches.add(
    new Match()
        .withKey("resource_name")
        .withValue("resource1")
);
body.withNotTagsAny(listbodyNotTagsAny);
body.withTagsAny(listbodyTagsAny);
body.withTags(listbodyTags);
body.withNotTags(listbodyNotTags);
body.withMatches(listbodyMatches);
body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("filter"));
body.withLimit("10");
body.withOffset("10");
request.withBody(body);
try {
    ListTransitIpsByTagsResponse response = client.listTransitIpsByTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
```

```
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 查询中转IP实例，其中，操作标识为count，按照条件将总条数返回。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class ListTransitIpsByTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListTransitIpsByTagsRequest request = new ListTransitIpsByTagsRequest();
        ListTagResourceInstancesRequestBody body = new ListTagResourceInstancesRequestBody();
        List<String> listNotTagsAnyValues = new ArrayList<>();
        listNotTagsAnyValues.add("value1");
        listNotTagsAnyValues.add("value2");
        List<Tags> listbodyNotTagsAny = new ArrayList<>();
        listbodyNotTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsAnyValues)
        );
        List<String> listTagsAnyValues = new ArrayList<>();
        listTagsAnyValues.add("value1");
        listTagsAnyValues.add("value2");
        List<Tags> listbodyTagsAny = new ArrayList<>();
        listbodyTagsAny.add(
            new Tags()
                .withKey("key1")
                .withValues(listTagsAnyValues)
        );
        List<String> listTagsValues = new ArrayList<>();
        listTagsValues.add("value1");
        listTagsValues.add("value2");
        List<String> listTagsValues1 = new ArrayList<>();
        listTagsValues1.add("value1");
        listTagsValues1.add("value2");
        List<Tags> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tags()
                .withKey("key1")
```



```
                .withValues(listTagsValues1)
        );
        listbodyTags.add(
            new Tags()
                .withKey("key2")
                .withValues(listTagsValues)
        );
        List<String> listNotTagsValues = new ArrayList<>();
        listNotTagsValues.add("value1");
        listNotTagsValues.add("*value2");
        List<Tags> listbodyNotTags = new ArrayList<>();
        listbodyNotTags.add(
            new Tags()
                .withKey("key1")
                .withValues(listNotTagsValues)
        );
        List<Match> listbodyMatches = new ArrayList<>();
        listbodyMatches.add(
            new Match()
                .withKey("resource_name")
                .withValue("resource1")
        );
        body.withNotTagsAny(listbodyNotTagsAny);
        body.withTagsAny(listbodyTagsAny);
        body.withTags(listbodyTags);
        body.withNotTags(listbodyNotTags);
        body.withMatches(listbodyMatches);
        body.withAction(ListTagResourceInstancesRequestBody.ActionEnum.fromValue("count"));
        request.withBody(body);
        try {
            ListTransitIpsByTagsResponse response = client.listTransitIpsByTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 查询中转IP实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

    client = NatClient.new_builder() \
```

```
    .with_credentials(credentials) \
    .with_region(NatRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ListTransitIpsByTagsRequest()
    listValuesNotTagsAny = [
        "value1",
        "value2"
    ]
    listNotTagsAnybody = [
        Tags(
            key="key1",
            values=listValuesNotTagsAny
        )
    ]
    listValuesTagsAny = [
        "value1",
        "value2"
    ]
    listTagsAnybody = [
        Tags(
            key="key1",
            values=listValuesTagsAny
        )
    ]
    listValuesTags = [
        "*value1",
        "value2"
    ]
    listTagsbody = [
        Tags(
            key="key1",
            values=listValuesTags
        )
    ]
    listValuesNotTags = [
        "*value1",
        "value2"
    ]
    listNotTagsbody = [
        Tags(
            key="key1",
            values=listValuesNotTags
        )
    ]
    listMatchesbody = [
        Match(
            key="resource_name",
            value="resource1"
        )
    ]
    request.body = ListTagResourceInstancesRequestBody(
        not_tags_any=listNotTagsAnybody,
        tags_any=listTagsAnybody,
        tags=listTagsbody,
        not_tags=listNotTagsbody,
        matches=listMatchesbody,
        action="filter",
        limit="10",
        offset="10"
    )
    response = client.list_transit_ips_by_tags(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 查询中转IP实例，其中，操作标识为count，按照条件将总条数返回。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTransitIpsByTagsRequest()
        listValuesNotTagsAny = [
            "value1",
            "value2"
        ]
        listNotTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesNotTagsAny
            )
        ]
        listValuesTagsAny = [
            "value1",
            "value2"
        ]
        listTagsAnybody = [
            Tags(
                key="key1",
                values=listValuesTagsAny
            )
        ]
        listValuesTags = [
            "value1",
            "value2"
        ]
        listValuesTags1 = [
            "value1",
            "value2"
        ]
        listTagsbody = [
            Tags(
                key="key1",
                values=listValuesTags1
            ),
            Tags(
                key="key2",
                values=listValuesTags
            )
        ]
        listValuesNotTags = [
            "value1",
            "*value2"
        ]
```

```
        listNotTagsbody = [
            Tags(
                key="key1",
                values=listValuesNotTags
            )
        ]
        listMatchesbody = [
            Match(
                key="resource_name",
                value="resource1"
            )
        ]
        request.body = ListTagResourceInstancesRequestBody(
            not_tags_any=listNotTagsAnybody,
            tags_any=listTagsAnybody,
            tags=listTagsbody,
            not_tags=listNotTagsbody,
            matches=listMatchesbody,
            action="count"
        )
        response = client.list_transit_ips_by_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 查询中转IP实例，其中，操作标识为filter，进行分页查询，查询记录数为10条。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTransitIpsByTagsRequest{}
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.Tags{
        {
```

```
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "*value1",
        "value2",
    }
    var listTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "*value1",
        "value2",
    }
    var listNotTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTags,
        },
    }
    var listMatchesbody = []model.Match{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    limitListTagResourceInstancesRequestBody:= "10"
    offsetListTagResourceInstancesRequestBody:= "10"
    request.Body = &model.ListTagResourceInstancesRequestBody{
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
        Action: model.GetListTagResourceInstancesRequestBodyActionEnum().FILTER,
        Limit: &limitListTagResourceInstancesRequestBody,
        Offset: &offsetListTagResourceInstancesRequestBody,
    }
    response, err := client.ListTransitIpsByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 查询中转IP实例，其中，操作标识为count，按照条件将总条数返回。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)
```

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTransitIpsByTagsRequest{}
    var listValuesNotTagsAny = []string{
        "value1",
        "value2",
    }
    var listNotTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTagsAny,
        },
    }
    var listValuesTagsAny = []string{
        "value1",
        "value2",
    }
    var listTagsAnybody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTagsAny,
        },
    }
    var listValuesTags = []string{
        "value1",
        "value2",
    }
    var listValuesTags1 = []string{
        "value1",
        "value2",
    }
    var listTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesTags1,
        },
        {
            Key: "key2",
            Values: listValuesTags,
        },
    }
    var listValuesNotTags = []string{
        "value1",
        "*value2",
    }
    var listNotTagsbody = []model.Tags{
        {
            Key: "key1",
            Values: listValuesNotTags,
```

```
        },
    }
    var listMatchesbody = []model.Match{
        {
            Key: "resource_name",
            Value: "resource1",
        },
    }
    request.Body = &model.ListTagResourceInstancesRequestBody{
        NotTagsAny: &listNotTagsAnybody,
        TagsAny: &listTagsAnybody,
        Tags: &listTagsbody,
        NotTags: &listNotTagsbody,
        Matches: &listMatchesbody,
        Action: model.GetListTagResourceInstancesRequestBodyActionEnum().COUNT,
    }
    response, err := client.ListTransitIpsByTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	• 查询操作成功。 • 示例1：action为count时的响应体 • 示例2：action为filter时的响应体

错误码

请参见**错误码**。

5.6.2 查询中转 IP 项目标签

功能介绍

- 查询租户在指定Project的所有中转IP标签集合。
- 标签管理服务需要能够列出当前租户全部已使用的中转IP标签集合，为打中转IP标签和过滤中转IP实例时提供标签联想功能。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/transit-ips/tags

表 5-143 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**

请求参数

表 5-144 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-145 响应 Body 参数

参数	参数类型	描述
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
tags	Array of **Tags** objects	标签。 数组长度：**1 - 10**

表 5-146 Tags

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。搜索时不对此参数做校验，key不能为空或者空字符串，不能为空格，校验和使用之前先trim 前后空格。 最小长度：**1** 最大长度：**128**
values	Array of strings	值列表。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/tags
```

响应示例

状态码： 200

查询操作成功。

```
{
  "request_id" : "36479272a29de0be0a8a8b294c02ab7a",
  "tags" : [ {
    "key" : "keys",
    "values" : [ "value" ]
  }, {
    "key" : "key3",
    "values" : [ "value3", "value33" ]
  }, {
    "key" : "key1",
    "values" : [ "value1" ]
  }, {
    "key" : "key2",
    "values" : [ "value2", "value22" ]
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;
```

```
public class ListTransitIpTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ListTransitIpTagsRequest request = new ListTransitIpTagsRequest();
        try {
            ListTransitIpTagsResponse response = client.listTransitIpTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTransitIpTagsRequest()
        response = client.list_transit_ip_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

5.6.3 查询中转 IP 标签

功能介绍

- 查询指定中转IP实例的标签信息。
- 标签管理服务需要使用该接口查询指定中转IP实例的全部标签数据。

调用方法

请参见**如何调用API**。

URI

GET /v3/{project_id}/transit-ips/{resource_id}/tags

表 5-147 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-148 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

状态码： 200

表 5-149 响应 Body 参数

参数	参数类型	描述
request_id	String	请求id。 最小长度：**1** 最大长度：**36**
tags	Array of **Tag** objects	标签。 数组长度：**1 - 10**

表 5-150 Tag

参数	参数类型	描述
key	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

请求示例

```
GET https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/b0399473-c352-4d0c-8ff4-cfde719cfde9/tags
```

响应示例

状态码： 200

查询操作成功。

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

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class ShowTransitIpTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTransitIpTagsRequest request = new ShowTransitIpTagsRequest();
        try {
            ShowTransitIpTagsResponse response = client.showTransitIpTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

```
    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询操作成功。

错误码

请参见**错误码**。

5.6.4 添加中转 IP 标签

功能介绍

- 一个中转IP上最多有10个标签。
- 此接口为幂等接口：
- 创建时，如果创建的标签已经存在（key相同），则覆盖。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/transit-ips/{resource_id}/tags

表 5-151 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-152 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-153 请求 Body 参数

参数	是否必选	参数类型	描述
tag	是	**Tag** object	标签。

表 5-154 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	是	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

无

请求示例

添加中转IP标签，其中，标签键为“key1”，对应的值为“value1”。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/56121618-fb0a-4a51-aff0-e2eb9cba4c73/tags

{
```

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

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

添加中转IP标签，其中，标签键为“key1”，对应的值为“value1”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class CreateTransitIpTagSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateTransitIpTagRequest request = new CreateTransitIpTagRequest();
        CreateResourceTagRequestBody body = new CreateResourceTagRequestBody();
        Tag tagbody = new Tag();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateTransitIpTagResponse response = client.createTransitIpTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
```

```
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

添加中转IP标签，其中，标签键为“key1”，对应的值为“value1”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = CreateTransitIpTagRequest()
        tagbody = Tag(
            key="key1",
            value="value1"
        )
        request.body = CreateResourceTagRequestBody(
            tag=tagbody
        )
        response = client.create_transit_ip_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

添加中转IP标签，其中，标签键为“key1”，对应的值为“value1”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTransitIpTagRequest{}
    tagbody := &model.Tag{
        Key: "key1",
        Value: "value1",
    }
    request.Body = &model.CreateResourceTagRequestBody{
        Tag: tagbody,
    }
    response, err := client.CreateTransitIpTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	添加操作成功。

错误码

请参见**错误码**。

5.6.5 批量添加删除中转 IP 标签

功能介绍

- 为指定中转IP实例批量添加或删除标签
- 标签管理服务需要使用该接口批量管理中转IP实例的标签。
- 一个中转IP上最多有10个标签。

接口约束

此接口为幂等接口：

- 创建时如果请求体中存在重复key则报错。
- 创建时，不允许设置重复key数据,如果数据库已存在该key，就覆盖value的值。
- 删除时，如果删除的标签不存在，默认处理成功，删除时不对标签字符集范围做校验。
- 删除时tags结构体不能缺失，key不能为空，或者空字符串。

调用方法

请参见**如何调用API**。

URI

POST /v3/{project_id}/transit-ips/{resource_id}/tags/action

表 5-155 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-156 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

表 5-157 请求 Body 参数

参数	是否必选	参数类型	描述
action	是	String	功能说明：操作标识。 取值范围： create（创建） delete（删除） 枚举值： • **create** • **delete**
tags	是	Array of **Tag** objects	标签列表。 数组长度：**1 - 10**

表 5-158 Tag

参数	是否必选	参数类型	描述
key	是	String	键。最大长度128个unicode字符。key不能为空。 最小长度：**1** 最大长度：**128**
value	是	String	值。每个值最大长度255个unicode字符。 最小长度：**0** 最大长度：**255**

响应参数

无

请求示例

- 批量添加中转IP标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
POST https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/56121618-fb0a-4a51-aff0-e2eb9cba4c73/tags/action

{
  "action" : "create",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

- 批量删除中转IP标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
POST  https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/56121618-fb0a-4a51-aff0-e2eb9cba4c73/tags/action

{
  "action" : "delete",
  "tags" : [ {
    "key" : "key1",
    "value" : "value1"
  }, {
    "key" : "key2",
    "value" : "value2"
  } ]
}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

- 批量添加中转IP标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeleteTransitIpTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
```



```
        BatchCreateDeleteTransitIpTagsRequest request = new BatchCreateDeleteTransitIpTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateDeleteTransitIpTagsResponse response =
client.batchCreateDeleteTransitIpTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 批量删除中转IP标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;

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

public class BatchCreateDeleteTransitIpTagsSolution {

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

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
```

```
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchCreateDeleteTransitIpTagsRequest request = new BatchCreateDeleteTransitIpTagsRequest();
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchCreateDeleteTransitIpTagsResponse response =
client.batchCreateDeleteTransitIpTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- 批量添加中转IP标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeleteTransitIpTagsRequest()
        listTagsbody = [
```

```
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_delete_transit_ip_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 批量删除中转IP标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = BatchCreateDeleteTransitIpTagsRequest()
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="delete"
        )
        response = client.batch_create_delete_transit_ip_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
        print(e.error_code)
        print(e.error_msg)
```

Go

- 批量添加中转IP标签，其中，操作标识为create，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeleteTransitIpTagsRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().CREATE,
    }
    response, err := client.BatchCreateDeleteTransitIpTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 批量删除中转IP标签，其中，操作标识为delete，标签列表中包含两个标签，其中第一个标签键为“key1”，对应的值为“value1”，第二个标签键为“key2”，对应的值为“value2”。

```
package main
```



```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateDeleteTransitIpTagsRequest{}
    var listTagsbody = []model.Tag{
        {
            Key: "key1",
            Value: "value1",
        },
        {
            Key: "key2",
            Value: "value2",
        },
    }
    request.Body = &model.BatchOperateResourceTagsRequestBody{
        Tags: listTagsbody,
        Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().DELETE,
    }
    response, err := client.BatchCreateDeleteTransitIpTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	批量添加或删除操作成功。

错误码

请参见**错误码**。

5.6.6 删除中转 IP 标签

功能介绍

- 幂等接口：
- 删除时，不对标签字符集做校验，调用接口前必须要做encodeURI，服务端需要对接口uri做decodeURI。删除的key不存在报404，key不能为空或者空字符串。

调用方法

请参见**如何调用API**。

URI

DELETE /v3/{project_id}/transit-ips/{resource_id}/tags/{key}

表 5-159 路径参数

参数	是否必选	参数类型	描述
key	是	String	标签key。 最小长度：**1** 最大长度：**128**
project_id	是	String	项目的ID。 最小长度：**1** 最大长度：**32**
resource_id	是	String	中转IP的ID。 最小长度：**36** 最大长度：**36**

请求参数

表 5-160 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 用户Token也就是调用获取用户Token获取请求认证接口的响应值，该接口是唯一不需要认证的接口。 请求响应成功后在响应消息头中包含的“X-Subject-Token”的值即为Token值。 最小长度：**1** 最大长度：**10240**

响应参数

无

请求示例

删除中转IP标签。

```
DELETE https://{Endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/transit-ips/56121618-fb0a-4a51-aff0-e2eb9cba4c73/tags/key1
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.nat.v2.region.NatRegion;
import com.huaweicloud.sdk.nat.v2.*;
import com.huaweicloud.sdk.nat.v2.model.*;


public class DeleteTransitIpTagSolution {

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

```
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        NatClient client = NatClient.newBuilder()
                .withCredential(auth)
                .withRegion(NatRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteTransitIpTagRequest request = new DeleteTransitIpTagRequest();
        try {
            DeleteTransitIpTagResponse response = client.deleteTransitIpTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdknat.v2.region.nat_region import NatRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdknat.v2 import *

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    nat "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/nat/v2/region"
)

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

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

    client := nat.NewNatClient(
        nat.NatClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

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

更多

更多编程语言的SDK代码示例，请参见**API Explorer**的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	删除操作成功。

错误码

请参见**错误码**。

6 应用示例

6.1 示例 1：创建公网 NAT 网关并配置 SNAT 规则

场景描述

本章节指导用户通过调用API来创建SNAT规则。API的调用方法请参见**如何调用API**。

前提条件

- 已创建VPC和子网，具体参见**创建云服务器所需要的VPC和子网**。
- 确认VPC下没有默认路由(“华北-北京四”区域无此限制)。
- 创建规则时，要求网关状态status = "ACTIVE"，要求网关管理员状态admin_state_up = True。
- 当您使用Token认证方式完成认证鉴权时，需要获取用户Token并在调用接口时增加“X-Auth-Token”到业务接口请求消息头中。Token认证，具体操作请参考**认证鉴权**。

说明

通过IAM服务获取到的Token有效期为24小时，需要使用同一个Token鉴权时，可以先将Token缓存，避免频繁调用。

操作步骤

步骤1 预置NAT网关。

1. 确定所用的VPC。
   - 查询VPC列表
     URI格式：GET /v1/{project_id}/vpcs
     详情请参见**查询VPC列表**。
   - 根据实际需要选择VPC并确认VPC下没有默认路由，然后记录VPC的id。
2. 确定所用VPC下的子网。
   - 查询所用的VPC下的子网列表
     URI格式：GET /v1/{project_id}/subnets?vpc_id={vpc_id}

详情请参见**查询子网列表**。

- 根据实际需要选择子网，并记录子网的id。

3. 创建公网NAT网关。
   - 接口相关信息

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

     详情请参见**创建公网NAT网关**。
   - 请求示例

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

     {endpoint}信息请从**地区和终端节点**获取。
   - 响应示例

```
{
   "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"
   }
}
```

4. 确定公网NAT网关创建成功，并且网关状态为active。

   接口相关信息

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

   详情请参见**查询指定的公网NAT网关详情**。
5. 记录新创建的NAT网关的id和对应的internal_network_id。

步骤2 确定待使用的弹性公网IP。

1. 查询弹性公网IP资源。

   接口相关信息

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

   详情请参见**查询弹性公网IP列表**。
2. 根据实际需要选择弹性EIP，并记录弹性EIP的id。

步骤3 创建SNAT规则。

- 接口相关信息

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

  接口约束及请求参数说明详情，请参见**创建SNAT规则**。
- 请求示例

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

  {endpoint}信息请从**地区和终端节点**获取。

  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"
```



```
    }
}
```

- 响应示例

```
{
    "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"
    }
}
```

步骤4 确认SNAT规则创建成功。

- 接口相关信息

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

  详情请参见**查询指定的SNAT规则详情**。

- 请求示例

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

  {endpoint}信息请从**地区和终端节点**获取。

- 响应示例

```
{
    "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"
    }
}
```

----结束

6.2 示例 2：创建公网 NAT 网关并配置 DNAT 规则

操作场景

本章节指导用户通过调用API来创建DNAT规则。API的调用方法请参见**如何调用API**。

前提条件

- 已创建VPC和子网，具体参见**创建云服务器所需要的VPC和子网**。
- 确认VPC下没有默认路由(“华北-北京四”区域无此限制)。

- 创建规则时，要求网关状态status = "ACTIVE"，要求网关管理员状态admin_state_up = True。
- 当您使用Token认证方式完成认证鉴权时，需要获取用户Token并在调用接口时增加“X-Auth-Token”到业务接口请求消息头中。Token认证，具体操作请参考**认证鉴权**。

说明

通过IAM服务获取到的Token有效期为24小时，需要使用同一个Token鉴权时，可以先将Token缓存，避免频繁调用。

操作步骤

步骤1 预置NAT网关。

1. 确定所用的VPC。
   - 查询VPC列表
     URI格式：GET /v1/{project_id}/vpcs
     详情请参见**查询VPC列表**。
   - 根据实际需要选择VPC并确认VPC下没有默认路由，然后记录VPC的id。
2. 确定所用VPC下的子网。
   - 查询所用的VPC下的子网列表
     URI格式：GET /v1/{project_id}/subnets?vpc_id={vpc_id}
     详情请参见**查询子网列表**。
   - 根据实际需要选择子网，并记录子网的id。
3. 创建公网NAT网关。
   - 接口相关信息
     URI格式：POST /v2/{project_id}/nat_gateways
     详情请参见**创建公网NAT网关**。
   - 请求示例
     POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/nat_gateways
     {endpoint}信息请从**地区和终端节点**获取。
   - 响应示例

```
{
   "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"
   }
}
```

4. 确定公网NAT网关创建成功，并且网关状态为active。
   接口相关信息
   URI格式：GET /v2/{project_id}/nat_gateways/{nat_gateway_id}
   详情请参见**查询指定的公网NAT网关详情**。
5. 记录新创建的公网NAT网关的id和对应的internal_network_id。

步骤2 确定待使用的弹性云服务器。

1. 查询云服务器详情列表。
   接口相关信息
   URI格式：GET https://{endpoint}/v1/{project_id}/cloudservers/detail
   详情请参见**查询云服务器详情列表**。
2. 根据实际需要选择弹性云服务器，并记录弹性云服务器的对应的网卡端口id。

步骤3 确定待使用的弹性公网IP。

1. 查询弹性公网IP资源。
   接口相关信息
   URI格式：GET /v1/{project_id}/publicips
   详情请参见**查询弹性公网IP列表**。
2. 根据实际需要选择弹性EIP，并记录弹性EIP的id。

步骤4 创建DNAT规则。

- 接口相关信息
  URI格式: POST /v2/{project_id}/dnat_rules
  接口约束及请求参数说明详情，请参见**创建DNAT规则**。
- 请求示例
  POST https://{Endpoint}/v2/27e25061336f4af590faeabeb7fcd9a3/dnat_rules
  {endpoint}信息请从**地区和终端节点**获取。
  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"
   }
}
```

- 响应示例

```
{
   "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"
   }
}
```

步骤5 确认DNAT规则创建成功。

- 接口相关信息

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

详情请参见**查询指定的DNAT规则详情**。

- 请求示例

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

  {endpoint}信息请从**地区和终端节点**获取。

- 响应示例

```
{
   "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"
   }
}
```

----结束

7 权限策略和授权项

7.1 策略及授权项说明

如果您需要对您所拥有的NAT网关（NAT Gateway）进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用NAT网关服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为**角色**和**策略**。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询NAT网关列表，那么这个IAM用户被授予的策略中必须包含允许“nat:natGateways:list”的授权项，该接口才能调用成功。

支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。

- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。

  说明

  “√”表示支持，“x”表示暂不支持。

NAT网关支持自定义策略授权项如下所示：

- **公网NAT网关**，包含NAT网关v2接口对应的授权项，如创建NAT网关、更新NAT网关、删除NAT网关等接口。
- **公网NAT网关SNAT规则**，包含SNAT规则v2接口对应的授权项，如创建SNAT规则、查询SNAT规则列表等接口。
- **公网NAT网关DNAT规则**，包含DNAT规则v2接口对应的授权项，如创建DNAT规则、查询DNAT规则列表等接口。

7.2 公网 NAT 网关

权限	对应API接口	授权项(Action)
创建公网NAT网关	POST /v2/{project_id}/nat_gateways	nat:natGateways:create
查询公网NAT网关列表	GET /v2/{project_id}/nat_gateways	nat:natGateways:list
查询公网NAT网关详情	GET /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:get
更新公网NAT网关	PUT /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:update
删除公网NAT网关	DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}	nat:natGateways:delete

7.3 公网 NAT 网关 SNAT 规则

权限	对应API接口	授权项(Action)
创建SNAT规则	POST /v2/{project_id}/snat_rules	nat:snatRules:create
查询SNAT规则列表	GET /v2/{project_id}/snat_rules	nat:snatRules:list
查询SNAT规则详情	GET /v2/{project_id}/snat_rules/{snat_rule_id}	nat:snatRules:get

权限	对应API接口	授权项(Action)
删除SNAT规则	DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}/snat_rules/{snat_rule_id}	nat:snatRules:delete
更新SNAT规则	PUT /v2/{project_id}/snat_rules/{snat_rule_id}	nat:snatRules:update

7.4 公网 NAT 网关 DNAT 规则

权限	对应API接口	授权项(Action)
创建DNAT规则	POST /v2/{project_id}/dnat_rules	nat:dnatRules:create
批量创建DNAT规则	POST /v2/{project_id}/dnat_rules/batch	nat:dnatRules:create
查询DNAT规则列表	GET /v2/{project_id}/dnat_rules	nat:dnatRules:list
查询DNAT规则详情	GET /v2/{project_id}/dnat_rules/{dnat_rule_id}	nat:dnatRules:get
删除DNAT规则	DELETE /v2/{project_id}/nat_gateways/{nat_gateway_id}/dnat_rules/{dnat_rule_id}	nat:dnatRules:delete
更新DNAT规则	PUT /v2/{project_id}/dnat_rules/{dnat_rule_id}	nat:dnatRules:update

8 附录

8.1 状态码

正常返回码	类型	说明
200	OK	GET和PUT操作正常返回。
201	Created	POST操作正常返回。
204	No Content	DELETE操作正常返回。

错误返回码	说明
400 Bad Request	服务器未能处理请求。
401 Unauthorized	被请求的页面需要用户名和密码。
403 Forbidden	对被请求页面的访问被禁止。
404 Not Found	服务器无法找到被请求的页面。
405 Method Not Allowed	请求中指定的方法不被允许。
406 Not Acceptable	服务器生成的响应无法被客户端所接受。
407 Proxy Authentication Required	用户必须首先使用代理服务器进行验证，这样请求才会被处理。
408 Request Timeout	请求超出了服务器的等待时间。
409 Conflict	由于冲突，请求无法被完成。
500 Internal Server Error	请求未完成，服务异常。
501 Not Implemented	请求未完成，服务器不支持所请求的功能。

错误返回码	说明
502 Bad Gateway	请求未完成，服务器从上游服务器收到一个无效的响应。
503 Service Unavailable	请求未完成，系统暂时异常。
504 Gateway Timeout	网关超时。

8.2 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见**API网关错误码**进行处理。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0001	Invalid value for created_at %(timestamp)s.	时间戳不合法	请检查输入的时间格式是否正确。
400	NAT.0002	Invalid parameters.	请求参数不合法	请检查输入的参数是否正确。
400	NAT.0006	Rule has not been deleted.	规则还没有被删除	请先排查一下是否有规则未删除导致删除网关失败。
400	NAT.0007	DB Error	数据库异常	请联系技术支持。
400	NAT.0008	Router %(router)s has no port for subnet %(subnet)s.	子网没有连接到虚拟路由器	请将子网添加到路由器接口。
400	NAT.0009	Resource %(res_type)s %(res)s is used by %(user_type)s %(user)s	资源被占用	请先排查一下该资源是否已被使用。
400	NAT.0010	Network %(network)s does not contain any IPv4 subnet	Network下没有Subnet	请先排查一下该network下是否添加subnet。
400	NAT.0012	The network %(network)s already has nat gateway.	Network已创建NAT网关	请选择一个未创建NAT网关的子网。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0014	Invalid input for description.+exceeds maximum length of 255.	参数description描述字符超过255个	请检查参数description描述字符个数。
400	NAT.0015	Invalid input for name.+exceeds maximum length of 255.	参数name字符超过255个	请检查参数name描述字符个数。
400	NAT.0016	Invalid input for spec. Reason: '*' is not in ['1', '2', '3', '4'].	参数spec不在（1,2,3,4）其中之一	请检查参数spec的值是否在（1,2,3，4）其中之一。
400	NAT.0017	Invalid input for router_id. Reason: '********-****-****-****-************' is not a valid UUID.	router_id不是合法的UUID	请检查输入的router_id是否正确。
400	NAT.0018	Invalid input for internal_network_id. Reason: '********-****-****-****-************' is not a valid UUID.	internal_network_id不是合法的UUID	请检查输入的internal_network_id是否正确。
400	NAT.0022	Either network_id or cidr must be specified. Both can not be specified at the same time	SNAT规则cidr和network冲突	请选择network_id和cidr两个输入一个即可。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0026	Floating IP ********-****-****-****-************ could not be found.	Floating_ip_id不存在	请检查输入的floating_ip_id是否正确。
400	NAT.0101	Lack of user authority. // request is null. // endpoint is empty. // resource type is invalid. // create natgateway requset is null. //update natgateway request is null // NatGateway id is invalid. //the enterprise project id is unsupported. //the enterprise project id in request is invalid. // request 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.	NAT网关请求错误	请根据错误提示排查一下错误原因或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0102	The system is busy. Please try again later.	系统繁忙，请稍后再试	系统繁忙，请稍后再试。
400	NAT.0103	NatGateway %(nat_gateway_id)s is not ACTIVE.	NAT网关未处于激活状态	请确认网关状态，如果长时间处于非“运行”状态，请联系技术支持。
400	NAT.0104	NatGateway %(nat_gateway_id)s is not UP. // NatGateway %(nat_gateway_id)s is frozen.can not update.	NAT网关处于冻结状态	请确认网关状态，网关可能因为欠费等原因被冻结，不允许进行更新操作。
400	NAT.0106	Concurrent conflict requests found	NAT网关并发操作冲突	请联系技术支持。
400	NAT.0107	Create NG Port failed.	NAT网关内部端口创建失败	NAT网关内部错误，请联系技术支持。
400	NAT.0108	NG Port %(port)s is unbound.	NAT网关内部端口未绑定	NAT网关内部错误，请联系技术支持。
400	NAT.0109	NatGateway does not support IPv6.	NAT网关不支持绑定IPv6类型的弹性公网IP	请绑定IPv4类型的弹性公网IP。
400	NAT.0110	Get Nat gateway host failed	选择网关节点错误	请联系技术支持。
400	NAT.0111	Get Nat gateway agent local_ip failed	获取网关节点的IP错误	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0112	Get RouteTable %(router_id)s failed.	获取VPC路由表错误	请联系技术支持。
400	NAT.0113	%(limit)s NAT gateways has been created to this VPC, no more is allowed	NAT网关数目超出规格	请选择其他VPC创建NAT网关或者删除该VPC下已创建的NAT网关。
400	NAT.0201	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.	SNAT规则参数错误	请确认输入的参数是否正确或联系技术支持。
400	NAT.0202	Either network_id or cidr must be specified.Both can not be specified at the same time	SNAT规则cidr和network冲突	请配置SNAT规则时不要同时指定Cidr和Network_id字段。
400	NAT.0203	cidr is invalid, make sure it's format is correct.	SNAT规则cidr不合法	请输入合法的CIDR，例如192.168.0.0/24。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0204	source_type and network_id is incompatible.	SNAT规则类型不合法	如果为VPC配置SNAT规则，Source_Type可不携带，如果携带，必须为0；
400	NAT.0205	cidr must be a subset of subnet's cidr.	VPC侧SNAT规则中的cidr必须是子网网段的子集	如果为VPC配置SNAT规则，CIDR必须是子网的子集。例如：子网为192.168.0.0/24,CIDR可填写192.168.0.0/25。
400	NAT.0206	cidr conflicts with subnet's cidr.	SNAT规则cidr与子网网段冲突	如果为专线配置SNAT规则，CIDR不能与VPC的子网冲突。
400	NAT.0207	cidr in the request conflicts with cidrs of existing rules.	SNAT网段冲突	请输入与现有SNAT规则不冲突的CIDR。
400	NAT.0208	Snat rule for network %(network)s exists.	规则已存在	请选择一个未配置SNAT规则的子网。
400	NAT.0210	Invalid input for floating_ip_id. Reason: \'%(fip)s\' is not a valid UUID. // Invalid value for public ip id.	SNAT规则的公网地址的ID不是合法的UUID	请输入合法的UUID。
400	NAT.0211	%(limit)s EIP has been associated to this SNAT rules's EIP pool, no more is allowed.	SNAT规则绑定的EIP个数超过上限	请保持弹性公网IP个数至上限以内。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0212	SNAT Rule %(rule)s Associated or disassociate EIP %(fip)s Failed."	SNAT规则绑定或解绑EIP失败	请联系技术支持。
400	NAT.0301	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 xxx is null in request body.	DNAT规则参数错误	请确认输入的参数是否正确或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0302	Dnat rule protocol % (protocol)s not supported.Only protocol values % (values)s and integer representations [6, 17, 0] are supported.	DNAT规则协议不合法	请配置合法的协议，范围在[6, 17, 0]中，对应TCP、UDP、ANY。
400	NAT.0303	Invalid value for port % (port)s	DNAT规则端口不合法	请配置合法的内网端口和外网端口，范围0-65535。
400	NAT.0304	The port_id, private_ip, internal port and protocol specified have been occupied.	DNAT规则内网信息冲突	请输入与现有DNAT规则不冲突的虚拟机Port ID或私网地址及私网端口。
400	NAT.0305	The floating ip, external port and protocol specified have been occupied.	DNAT规则外网信息冲突	请输入与现有DNAT规则不冲突的浮动IP的ID、外网端口、协议。
400	NAT.0306	The external port equals 0 and internal port equals 0 and protocol equals any must satisfied at the same time.	DNAT规则AllPort类型请求信息错误	私网端口和外网端口都为0及协议为any必须同时配置才生效。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0307	The port_id already existing dnat allport rules or dnat_rules, can no longer create dnat rules or dnat allport rules.	DNAT规则中port_id与已有Dnat规则冲突	虚拟机Port ID与已有DNAT规则冲突，请更换虚拟机Port ID创建或更新dnat规则。
400	NAT.0308	The private_ip already existing dnat allport rules or dnat rules, can no longer create dnat rules or dnat allport rules.	DNAT规则中私网IP与已有DNAT规则冲突	私网地址与已有DNAT规则冲突，请更换私网地址创建或更新dnat规则。
400	NAT.0309	%(limit)s DNAT rules has been associated to this NAT Gateway, no more is allowed	DNAT规则条数超出规格	同一nat网关的DNAT规则条数超出上限。
400	NAT.0310	The port_id and private_ip values are both empty, at least one value is not empty.	DNAT规则缺少参数	虚拟机Port ID和私网地址都没有配置，请确认配置。
400	NAT.0311	The private ip address is not legal.	DNAT规则私网IP不合法	请配置合法的私网地址。
400	NAT.0312	The virtual IP address is not supported.	不支持虚拟IP地址	私网地址不支持配置为虚拟IP地址，请配置合法的私网地址。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0313	%(limit)s DNAT rules has been associated to this Floating IP, no more is allowed	DNAT规则条数超出规格	绑定同一浮动IP的DNAT规则条数超出上限。
400	NAT.0314	batch create dnat rules max limit: %(limit)s	批量添加DNAT规则条数超出规格	请减少DNAT规则条数进行批量添加。
400	NAT.0315	Port %(port)s is not a valid port.	DNAT规则虚拟机Port ID不合法	请配置合法的虚拟机Port ID。
400	NAT.0316	Vtep_ip is Null.	VtepIp参数不能为空	请删除重建DNAT规则或联系技术支持。
400	NAT.0317	The port_id and private_ip exist at the same time and value is not empty, but at least one value is empty.	DNAT规则存在互斥参数	虚拟机Port ID和私网地址不能同时配置，请确认配置。
400	NAT.0318	DNAT rule is frozen, can no longer update.	DNAT规则被冻结，无法更新	请检查DNAT规则绑定的floating_ip是否处于欠费状态或用户账户是否处于欠费状态。
400	NAT.0401	Floating Ip %(fip)s is freezed.	EIP处于冻结状态	请重新选择未被冻结的弹性公网IP地址。
400	NAT.0402	Floating Ip %(fip)s has associated with port %(port)s.	EIP关联限制（已关联到另一个端口）	请选择未绑定到其他对象的弹性公网IP地址，例如弹性公网IP地址绑定到了ECS，则不能再绑定到NAT网关。
400	NAT.0403	There is a duplicate EIP %(fips)s in SNAT rule.	公网IP已被SNAT规则占用，请重新绑定	该弹性公网IP已被SNAT规则绑定，请重新绑定其他弹性公网IP。

状态码	错误码	错误信息	描述	处理措施
400	NAT.0404	Floating Ip %(fip)s has used by nat gateway %(nat_gateway)s.	EIP关联限制（已关联到另一个NAT网关实例）	弹性公网IP已经绑定到其他NAT网关，请重新选择。
400	NAT.0405	Floating Ip %(fip)s has been occupied.	EIP已被占用	弹性公网IP已经绑定到其他NAT网关，请重新选择。
400	NAT.0407	Floating Ip %(fip)s is used by other rules	EIP关联限制(已关联到另一条规则)	弹性公网IP被其他规则使用，请重新选择。
400	NAT.0408	Floating Ip %(fip)s can not be associated with both DNAT rule and DNAT all port rule.	EIP关联限制(不能同时关联dnat和dnat all port)	弹性公网IP不能同时绑定到DNAT和DNAT all port规则，请重新选择。
400	NAT.0409	Floating Ip %(fip)s can not be associated with both SNAT rule and DNAT all port rule.	公网IP关联限制(SNAT和DNAT all port不能共用)	SNAT和DNAT all port不能共用一条弹性公网IP，请重新选择。
400	NAT.0410	Invalid value of the FloatIP.	公网IP无效	请检查输入的floating_ip是否有效
400	VPC.0002	Available zone Name is null.	可用区为空	请确认创建子网的请求体中availability_zone字段是否为空。
400	VPC.0004	VPC does not active, please try later.	VPC状态异常	请稍后重试或联系技术支持。
400	VPC.0007	urlTenantId is not equal tokenTenantId	url里的tenant_id和token中解析到的tenant_id不一致	请联系技术支持。
400	VPC.0011	EnterpriseProjectId is invalid	企业项目id非法	请输入合法的企业项目id。

状态码	错误码	错误信息	描述	处理措施
400	VPC.0014	This enterpriseProject status is disable.	企业项目不可用	请更换其他可用企业项目id。
400	VPC.2000	Lack of user authority. // request is null. // endpoint is empty. // resource type is invalid. // create natgateway requset is null. //update natgateway request is null.	NAT网关请求错误	请联系技术支持。
400	VPC.2001	NatGateway id is invalid. //the enterprise project id in request is invalid. // request 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.	NAT网关参数错误	请确认输入的参数是否正确或联系技术支持。
400	VPC.2002	Invalid parameters.	请求参数不合法	请检查输入的参数是否正确。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2004	NatGateway %(nat_gateway_id)s is not ACTIVE.	NAT网关未处于激活状态	请确认网关状态，如果长时间处于非“运行”状态，请联系技术支持。
400	VPC.2005	NatGateway %(nat_gateway_id)s is not UP.	NAT网关未处于UP状态	请检查网关是否因为欠费等原因被冻结。
400	VPC.2006	NatGateway %(nat_gateway_id)s is frozen.can not update	NAT网关处于冻结状态	请检查网关是否因为欠费等原因被冻结，不允许进行更新操作。
400	VPC.2007	NatGateway %(nat_gateway_id)s does not exist.	NAT网关不存在	请检查输入的NAT网关是否正确。
400	VPC.2008	Network %(network)s does not contain any IPv4 subnet	Network下没有Subnet	请联系技术支持。
400	VPC.2009	Network %(network_id)s does not exist.	Network不存在	子网不存在，请输入合法的子网。
400	VPC.2010	The router %(router_id)s has default route.	默认路由已存在	路由器下已存在默认路由，请先删除默认路由再创建NAT网关。
400	VPC.2011	The router %(router_id)s does not exist.	Router不存在	路由器不存在，请检查输入的路由器是否正确。
400	VPC.2012	The router %(router_id)s already has nat gateway.	VPC下已存在nat网关	VPC下已存在NAT网关，请选择其他VPC。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2013	Router %(router)s has no port for subnet %(subnet)s.	子网没有连接到虚拟路由器	请将子网添加到路由器接口。
400	VPC.2014	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.	SNAT规则参数错误	请确认输入的参数是否正确或联系技术支持。
400	VPC.2016	Rule has not been deleted.	规则还没有被删除	请联系技术支持。
400	VPC.2018	Snat rule for network %(network)s exists.	规则已存在	请选择一个未配置SNAT规则的子网。
400	VPC.2019	Resource %(res_type)s %(res)s is used by %(user_type)s %(user)s	资源被占用	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2020	get dnatRules error limit is invalid. //get dnatrules error marker is invalid. // endpoint is empty. // DnatRule id invalid. // Request is null. // DnatRule id invalid. // DnatRule natGatewayId id invalid.	DNAT规则参数错误	请确认输入的参数是否正确或联系技术支持。
400	VPC.2022	Port %(port)s is not a valid port.	DNAT规则虚拟机Port ID不合法	请配置合法的虚拟机Port ID。
400	VPC.2023	The port_id, private_ip, internal port and protocol specified have been occupied.	DNAT规则内网信息冲突	请输入与现有DNAT规则不冲突的虚拟机Port ID或私网地址及私网端口。
400	VPC.2024	The floating ip, external port and protocol specified have been occupied.	DNAT规则外网信息冲突	请输入与现有DNAT规则不冲突的浮动IP的ID、外网端口、协议。
400	VPC.2026	%(limit)s DNAT rules has been associated to this Floating IP, no more is allowed	绑定同一浮动IP的DNAT规则条数超出上限	请减少DNAT规则条数。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2027	The port_id already existing dnat allport rules or dnat_rules, can no longer create dnat rules or dnat allport rules.	DNAT规则中port_id与已有Dnat规则冲突	虚拟机Port ID与已有DNAT规则冲突，请更换虚拟机Port ID创建或更新dnat规则。
400	VPC.2028	The private_ip already existing dnat allport rules or dnat rules, can no longer create dnat rules or dnat allport rules.	DNAT规则中私网IP与已有Dnat规则冲突	私网地址与已有DNAT规则冲突，请更换私网地址创建或更新DNAT规则。
400	VPC.2029	DNAT rule is frozen, can no longer update.	DNAT规则被冻结，无法更新	请检查DNAT规则绑定的floating_ip是否处于欠费状态或用户账户是否处于欠费状态。
400	VPC.2030	The system is busy. Please try again later.	系统繁忙，请稍后再试	系统繁忙，请稍后再试。
400	VPC.2031	Either network_id or cidr must be specified.Both can not be specified at the same time	SNAT规则cidr和network冲突	请配置SNAT规则时不要同时指定Cidr和Network_id字段。
400	VPC.2032	cidr is invalid, make sure it's format is correct.	SNAT规则cidr不合法	请输入合法的CIDR，例如192.168.0.0/24。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2033	source_type and network_id is incompatible.	SNAT规则类型不合法	如果为VPC配置SNAT规则，Source_Type可不携带，如果携带，必须为0；如果为专线侧配置SNAT规则，Source_type必须为1。
400	VPC.2034	cidr must be a subset of subnet's cidr.	VPC侧SNAT规则中的cidr必须是子网网段的子集	如果为VPC配置SNAT规则，CIDR必须是子网的子集。例如：子网为192.168.0.0/24,CIDR可填写192.168.0.0/25。
400	VPC.2035	cidr conflicts with subnet's cidr.	SNAT规则cidr与子网网段冲突	如果为专线配置SNAT规则，CIDR不能与VPC的子网冲突。
400	VPC.2036	cidr in the request conflicts with cidrs of existing rules.	SNAT网段冲突	请输入与现有SNAT规则不冲突的CIDR。
400	VPC.2037	The virtual IP address is not supported.	不支持虚拟IP地址	私网地址不支持配置为虚拟IP地址，请配置合法的私网地址。
400	VPC.2038	%(limit)s DNAT rules has been associated to this NAT Gateway, no more is allowed	DNAT规则条数超出规格	请删除一些DNAT规则。
400	VPC.2039	%(limit)s EIP has been associated to this SNAT rules's EIP pool, no more is allowed.	绑定到SNAT规则的EIP个数超过上限	请减少弹性公网IP数量。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2040	Invalid value for public ip id.	SNAT规则的公网地址的ID不可以是空串	请输入合法的UUID。
400	VPC.2042	There is a duplicate EIP %(fips)s in SNAT rule.	EIP已被SNAT规则使用	该弹性公网IP已被SNAT规则绑定，请重新绑定其他弹性公网IP。
400	VPC.2043	Floating Ip %(fip)s is used by other rules	EIP关联限制(已关联到另一条规则)	请重新选择EIP。
400	VPC.2044	Invalid input for floating_ip_id. Reason: \'%(fip)s\' is not a valid UUID.	SNAT规则的公网地址的ID不是合法的UUID	请输入合法的UUID。
400	VPC.2045	Get Nat gateway host failed	选择网关节点错误	请联系技术支持。
400	VPC.2046	Get Nat gateway agent local_ip failed	获取网关节点的IP错误	请联系技术支持。
400	VPC.2047	Get RouteTable %(router_id)s failed.	获取VPC路由表错误	请联系技术支持。
400	VPC.2048	Invalid value for created_at %(timestamp)s.	时间戳不合法	请检查输入的时间格式是否正确。
400	VPC.2049	DB Error	数据库异常	请联系技术支持。
400	VPC.2050	Concurrent conflict requests found	NAT网关并发操作冲突	请联系技术支持。
400	VPC.2051	Create NG Port failed.	NAT网关内部端口创建失败	NAT网关内部错误，请联系技术支持。
400	VPC.2052	NG Port %(port)s is unbound.	NAT网关内部端口未绑定	NAT网关内部错误，请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2053	NatGateway does not support IPv6.	NAT网关不支持IPv6	请绑定IPv4类型的弹性公网IP。
400	VPC.2054	Dnat rule protocol %(protocol)s not supported.Only protocol values %(values)s and integer representations [6, 17, 0] are supported.	DNAT规则协议不合法	请配置合法的协议，范围在[6, 17, 0]中，对应TCP/UDP/ANY。
400	VPC.2055	The port_id and private_ip exist at the same time and value is not empty, but at least one value is empty.	DNAT规则存在互斥参数	虚拟机Port ID和私网地址不能同时配置，请确认配置。
400	VPC.2056	The port_id and private_ip values are both empty, at least one value is not empty.	DNAT规则缺少参数	虚拟机Port ID和私网地址都没有配置，请确认配置。
400	VPC.2057	batch create dnat rules max limit: %(limit)s	批量创建DNAT规则条数超出规格	请减少批量创建DNAT规则条数。
400	VPC.2058	Vtep_ip is Null.	VtepIp参数不能为空	请联系技术支持
400	VPC.2059	Floating Ip %(fip)s is freezed.	EIP处于冻结状态	请重新选择未被冻结的弹性公网IP地址

状态码	错误码	错误信息	描述	处理措施
400	VPC.2060	Floating Ip %(fip)s has associated with port %(port)s.	EIP关联限制（已关联到另一个端口）	请选择未绑定到其他对象的弹性公网IP地址，例如弹性公网IP地址绑定到了ECS，则不能再绑定到NAT网关
400	VPC.2061	Floating Ip %(fip)s has used by nat gateway %(nat_gateway)s.	EIP关联限制（已关联到另一个Nat网关实例）	请重新选择EIP。
400	VPC.2062	Floating Ip %(fip)s has been occupied.	EIP已被占用	请重新选择EIP。
400	VPC.2069	Invalid value for port %(port)s	DNAT规则端口不合法	请配置合法的内网端口和外网端口，范围0-65535。
400	VPC.2070	The external port equals 0 and internal port equals 0 and protocol equals any must satisfied at the same time.	DNAT规则AllPort类型请求信息错误	请同时配置私网端口和外网端口都为0及协议为any。
400	VPC.2071	The private ip address is not legal.	DNAT规则私网IP不合法	请配置合法的私网地址。
400	VPC.2073	Floating Ip %(fip)s can not be associated with both DNAT rule and DNAT all port rule.	EIP关联限制(不能同时关联DNAT和DNAT all port)	请重新选择EIP。
400	VPC.2074	Floating Ip %(fip)s can not be associated with both SNAT rule and DNAT all port rule.	EIP关联限制(不能同时关联SNAT和DNAT all port)	请重新选择EIP。

状态码	错误码	错误信息	描述	处理措施
400	VPC.2075	Enter a maximum of 255 characters.	描述字符数超过255个	请检查输入的描述字符个数
400/404	NAT.0105	NatGateway %(nat_gateway_id)s does not exist.	NAT网关不存在(http状态码400对应的是删除网关时网关不存在，404对应创建、查询时网关不存在)	请检查该nat_gateway_id是否存在。
401	NAT.0025	Token is expired.	Token过期	请检查输入的token是否在有效期。
401	VPC.0008	Invalid token in the header.	token非法	请确认请求头中的token是否合法。
401	VPC.0009	real-name authentication fail.	实名认证失败	请联系技术支持。
403	VPC.0010	Rules on xx by ** disallowed by policy	调用底层权限不足	请赋予正确的细粒度权限。
403	VPC.2201	Policy doesn't allow <x:x:x> to be performed	细粒度权限不足	请赋予正确的细粒度权限。
403	VPC.2701	Token not allowed to do this action.	无权操作，或账户余额不足	请确认账户是否余额不足或被冻结。
404	NAT.0004	The router %(router_id)s does not exist.	Router不存在	路由器不存在，请检查输入的路由器是否正确。
404	NAT.0005	Network %(network_id)s does not exist.	Network不存在	子网不存在，请输入合法的子网。

状态码	错误码	错误信息	描述	处理措施
404	NAT.0013	Router % (router)s for the specified NAT gateway could not be found.	没有找到指定NAT网关的路由	请为指定NAT网关创建路由。
404	NAT.0019	Network ********-****-****-****-************ could not be found.	network_id不存在	请检查输入的internal_network_id是否存在。
404	NAT.0020	Specifying 'tenant_id' other than authenticated tenant in request requires admin privileges	tenant_id为空或不存在	请检查输入的tenant_id是否存在。
404	NAT.0021	Invalid input for nat_gateway_id. Reason: '********-****-****-****-************'is not a valid UUID.	Nat_gateway_id为空或不存在	请检查输入的nat_gateway_id是否存在。
404	NAT.0023	Port '********-****-****-****-************' could not be found.	Port_id不存在	请检查输入的port_id是否存在。
404	NAT.0024	Invalid input for floating_ip_id. Reason: '********-****-****-****-************'is not a valid UUID.	Floating_ip_id不存在、为空或者非法	请检查输入的floating_ip_id是否正确。

状态码	错误码	错误信息	描述	处理措施
404	NAT.0209	No Snat Rule exist with id %(id)s	SNAT规则不存在	请检查SNAT规则的id是否存在。
404	NAT.0319	No Dnat Rule exist with id %(id)s	DNAT规则不存在	请联系技术支持。
404	VPC.0003	VPC does not exit.	VPC不存在	请确认VPC的id是否填写正确或该租户下是否确实存在该VPC。
400	NAT.1010	Request parameter Json parsing failed %s.	请求参数Json解析失败	请检查请求参数格式是否正确。
400	NAT.1015	Tags parameter is illegal.	tag参数不合法	请检查tag参数是否合法。
400	NAT.1016	The number of tags exceeds the limit %s.	tag数量超过最大约束	请设置不超过%s个tag。
400	NAT.1022	Account is restricted, operation is forbidden.	账号受限不允许创建私网NAT网关	请查看账号是否处于受限状态并联系技术支持。
400	NAT.1023	Account is suspended, operation is forbidden.	账号冻结不允许创建私网NAT网关	请查看账号是否处于冻结状态并联系技术支持。
400	NAT.1101	Transit subnet can not be create with this network %s. //External subnet can not be create with this network %s.	中转IP所在子网不可用	请联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	NAT.1201	%s downlink vpc has been associated to this private nat gateway, no more is allowed.	downlink_vpc的数量不允许超过最大约束	请输入最多1个downlink_vpc。
400	NAT.1202	There are one or more rules still in use on the gateway %s, can not be deleted.	私网NAT网关存在规则因此不允许被删除	请删除私网NAT网关的所有规则后再删除私网NAT网关。
400	NAT.1204	This vpc at most have %s private nat gateway.	VPC下私网NAT网关数量已达最大约束	请使用其他VPC或删除该VPC下的其他私网NAT网关。
400	NAT.1206	Update (%s) spec to (%s) are the same as the original.	不能使用相同的规格更新私网NAT网关	请使用和原始规格不同的规格更新私网NAT网关。
400	NAT.1207	Gateway %s already has %d rules, the maximum number of rule for spec %s is %d. Downgrade is forbidden. //Add this gateway rule over rule max %d.	私网NAT网关当前规则数已超过网关更新的规格的最大约束时，不允许更新网关规格。私网NAT网关规则数已达规格最大约束，不允许新建规则。	请确保私网NAT网关当前的规则数不会超过更新规格后的规格最大约束。请删除一些规则或扩大私网NAT网关的规格。

状态码	错误码	错误信息	描述	处理措施
400	NAT.1208	Gateway %s is frozen, create operation is forbidden. // Gateway %s is frozen, delete operation is forbidden. // Gateway %s is frozen, update operation is forbidden.	私网NAT网关已被冻结时禁止相关操作。	请检查私网NAT网关是否处于冻结状态并联系技术支持。
400	NAT.1306	dnat Parameters entered are illegal, protocol:any should be entered with internal_service_port:0 and transit_service_port:0 together. // dnat Parameters entered are illegal, protocol:any should be entered with internal_service_port:0 and external_service_port:0 together.	DNAT规则请求体中的参数不合法：protocol为any、internal_service_port为0、transit_service_port为0需同时配置。	请使用合法参数进行配置。

状态码	错误码	错误信息	描述	处理措施
400	NAT.1311	The networkInterfaceId(port_id) and private_ip_address(fixed_ip_address) exist at the same time or both are empty, but at least one value is empty.	network_interface_id和private_ip_address不能同时存在或同时为空，需指定其一	请指定network_interface_id或private_ip_address。
400	NAT.1404	No more IP addresses available on subnet %s.	子网内已不存在可用IP地址	请联系技术支持。
400	NAT.1405	IP address %s is not a valid IP for the subnet.	IP地址不是子网的有效IP	请联系技术支持。
400	NAT.1501	Either virSubnet(network) or cidr must be specified." + "Both can not be specified at the same time.	virsubnet_id和cidr参数二选一	请指定virsubnet_id或cidr。
400	NAT.1502	Cidr is invalid, make sure it's format is correct.	cidr格式不合法	请输入合法的cidr，例如192.168.0.0/24。

状态码	错误码	错误信息	描述	处理措施
400	NAT.1506	%s transit ip has been associated to this SNAT rules's transit ip pool, no more is allowed. //%s external ip has been associated to this SNAT rules's external ip pool, no more is allowed.	SNAT规则的中转IP数量超过了最大约束	请输入最多1个中转IP。
400	NAT.1507	'transit_ip_id' attribute value should be 'uuid' type! //'external_ip_id' attribute value should be 'uuid' type!	transit_ip_id不合法	请输入UUID格式的transit_ip_id。
404	NAT.1002	%s %s could not be found.	资源不存在	请查询资源是否存在或联系技术支持。
404	NAT.1003	VirSubnet %s could not be found in vpc for gateway %s.	SNAT规则后端子网不在私网NAT网关实例所属的VPC	请确保SNAT规则后端子网属于私网NAT网关实例所属的VPC。
404	NAT.1009	Port %s information is missing.	Port缺少信息	请联系技术支持。
409	NAT.1303	Transit ip %s is in this vpc, not support to specified. // External ip %s is in this vpc, not support to specified.	中转IP所属VPC与私网NAT网关所属VPC不能相同	请使用与私网NAT网关所属VPC不同的中转IP。

状态码	错误码	错误信息	描述	处理措施
409	NAT.1304	Port %s is not in this vpc, not support to specified.	DNAT规则后端的Port必须在私网NAT网关所属的VPC	请确保DNAT规则后端的Port存在于私网NAT网关所属的VPC。
409	NAT.1305	Transit ip %s is used by exist dnat rule. // External ip %s is used by exist dnat rule.	中转IP已被DNAT规则使用	请使用没有被DNAT规则使用的中转IP。
409	NAT.1307	Internal parameters entered conflict with exist dnat rules. // PrivateIp(FixedIp) %s is used by exist dnat rule.	DNAT规则内网信息冲突	请使用与已有DNAT规则不冲突的Port/私网IP、后端端口、协议。
409	NAT.1308	Transit ip %s is used by exist snat rule. // External ip %s is used by exist snat rule.	中转IP已被SNAT规则使用	请使用没有被SNAT规则使用的中转IP。
409	NAT.1309	Port %s is used by exist dnat rule.	Port已被其他协议类型的DNAT规则使用	请使用没有被其他协议类型的DNAT规则使用的Port。
409	NAT.1310	Transit parameters entered conflict with exist dnat rules. // External parameters entered conflict with exist dnat rules.	DNAT规则外网信息冲突	请使用与已有DNAT规则不冲突的中转IP、端口、协议。

状态码	错误码	错误信息	描述	处理措施
409	NAT.1403	Unable to complete operation for subnet %s. The IP address %s is in use.	IP地址已被使用	请联系技术支持。
409	NAT.1406	Transit ip %s has used by private nat gateway %s. //External ip %s has used by private nat gateway %s.	中转IP已被其他私网NAT网关使用	请使用未被其他私网NAT网关使用的中转IP。
409	NAT.1409	Transit ip %s is used by rules. // External ip %s is used by rules.	中转IP被规则使用	请确认中转IP不被其他规则使用后再进行删除。
409	NAT.1410	Transit ip %s is used by dnat rules of other protocols. // External ip %s is used by dnat rules of other protocols.	中转IP被其他协议类型的DNAT规则使用	请使用不被其他协议类型的DNAT规则使用的中转IP。
409	NAT.1503	Snat rule for network %s exists.	子网已被其他SNAT规则使用	请选择一个未配置SNAT规则的子网。
409	NAT.1505	Snat rule for cidr %s exists.	cidr已被其他SNAT规则使用	请输入与现有SNAT规则不相同的cidr。
500	NAT.1001	Internal Server Error.	内部服务错误	请联系技术支持。
500	NAT.1004	Create Port Failed with subnet %s.	子网内创建Port失败	请联系技术支持。
500	NAT.1005	Delete Port %s Failed.	删除Port失败	请联系技术支持。

8.3 获取项目 ID

操作场景

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。有如下两种获取方式：

- **调用API获取项目ID**
- **从控制台获取项目ID**

调用 API 获取项目 ID

项目ID可以通过调用**查询指定条件下的项目列表**API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从**地区和终端节点**获取。接口的认证鉴权请参见**认证鉴权**。

响应示例如下，其中projects下的“id”即为项目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"
    }
}
```

从控制台获取项目 ID

从控制台获取项目ID的步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。
   在“API凭证”页面的项目列表中查看项目ID。

图 8-1 查看项目 ID

8.4 资源状态说明

表 8-1 资源状态说明

资源状态	说明
ACTIVE	当前资源状态正常
PENDING_CREATE	当前资源正在创建
PENDING_UPDATE	当前资源正在更新
PENDING_DELETE	当前资源正在删除
EIP_FREEZED	当前资源存在EIP被冻结
INACTIVE	当前资源状态异常

9 历史 API

9.1 API v2.0

9.1.1 公网 NAT 网关

9.1.1.1 创建公网 NAT 网关

功能介绍

创建公网NAT网关实例。

URI

POST /v2.0/nat_gateways

请求消息

请求参数如表9-1所示。

表 9-1 请求参数

参数	是否必选	参数类型	描述
nat_gateway	是	Object	公网NAT网关对象。详见表9-2。

表 9-2 nat_gateway 字段说明

参数	是否必选	参数类型	描述
tenant_id	否	String	项目的ID。

参数	是否必选	参数类型	描述
name	是	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	否	String	公网NAT网关的描述，长度限制为255。
spec	是	String	公网NAT网关的规格。 取值为： • “1”：小型，SNAT最大连接数10000 • “2”：中型，SNAT最大连接数50000 • “3”：大型，SNAT最大连接数200000 • “4”：超大型，SNAT最大连接数1000000
router_id	是	String	VPC的id。
internal_network_id	是	String	公网NAT网关下行口（DVR的下一跳）所属的network id。

响应消息

响应参数如表9-3所示。

表 9-3 响应参数

参数	参数类型	描述
nat_gateway	Object	nat_gateway对象。详见表9-4。

表 9-4 nat_gateway 字段说明

参数	参数类型	描述
id	String	公网NAT网关的id。
tenant_id	String	项目的ID。
name	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	String	公网NAT网关的描述，长度限制为255。

参数	参数类型	描述
spec	String	公网NAT网关的规格。 取值为： ● “1”：小型，SNAT最大连接数10000 ● “2”：中型，SNAT最大连接数50000 ● “3”：大型，SNAT最大连接数200000 ● “4”：超大型，SNAT最大连接数1000000
router_id	String	路由器的ID。
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。
status	String	● 功能说明：公网NAT网关的状态。 ● 取值范围：资源状态说明。
admin_state_up	Boolean	● 解冻/冻结状态。 ● 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	公网NAT网关的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求示例

```
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"
   }
}
```

- 响应示例

```
{
   "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"
   }
}
```

状态码

请参考状态码。

9.1.1.2 查询公网 NAT 网关列表

功能介绍

查询公网NAT网关列表。如无特殊说明，匹配规则为精确匹配。

URI

GET /v2.0/nat_gateways

说明

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。支持参数说明中所有非必选参数过滤，请参考请求样例。

表 9-5 参数说明

参数	是否必选	参数类型	描述
id	否	String	公网NAT网关的id。
limit	否	Integer	每页返回的个数。
tenant_id	否	String	项目的ID。
name	否	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	否	String	公网NAT网关的描述，长度限制为255。
spec	否	String	公网NAT网关的规格。 取值为： • “1”：小型，SNAT最大连接数10000 • “2”：中型，SNAT最大连接数50000 • “3”：大型，SNAT最大连接数200000 • “4”：超大型，SNAT最大连接数1000000
router_id	否	String	路由器的ID。
internal_network_id	否	String	公网NAT网关下行口（DVR的下一跳）所属的network id。
status	否	String	• 功能说明：公网NAT网关的状态。 • 取值范围：资源状态说明。

参数	是否必选	参数类型	描述
admin_state_up	否	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	否	String	公网NAT网关的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

请求消息

无

响应消息

响应参数如表9-6所示。

表 9-6 响应参数

参数	参数类型	描述
nat_gateways	List(nat_gateway)	nat_gateway对象列表。详见表9-7。

表 9-7 nat_gateway 字段说明

参数	参数类型	描述
id	String	公网NAT网关的id。
tenant_id	String	项目的ID。
name	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	String	公网NAT网关的描述，长度限制为255。

参数	参数类型	描述
spec	String	公网NAT网关的规格。 取值为： ● “1”：小型，SNAT最大连接数10000 ● “2”：中型，SNAT最大连接数50000 ● “3”：大型，SNAT最大连接数200000 ● “4”：超大型，SNAT最大连接数1000000
router_id	String	路由器的ID。
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。
status	String	● 功能说明：公网NAT网关的状态。 ● 取值范围：资源状态说明。
admin_state_up	Boolean	● 解冻/冻结状态。 ● 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	公网NAT网关的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例
  ```
  GET https://{Endpoint}/v2.0/nat_gateways?limit=10
  ```
- 响应样例
  ```
  {
      "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"
        }
    ]
}
```

状态码

请参考**状态码**。

9.1.1.3 查询指定的公网 NAT 网关详情

功能介绍

查询指定的公网NAT网关详情。

URI

GET /v2.0/nat_gateways/{nat_gateway_id}

表 9-8 参数说明

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	所属公网NAT网关的id。

请求消息

无

响应消息

响应参数如**表9-9**所示。

表 9-9 响应参数

参数	参数类型	描述
nat_gateway	Object	nat_gateway对象。详见**表9-10**。

表 9-10 nat_gateway 字段说明

参数	参数类型	描述
id	String	公网NAT网关的id。
tenant_id	String	项目的ID。

参数	参数类型	描述
name	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	String	公网NAT网关的描述，长度限制为255。
spec	String	公网NAT网关的规格。 取值为： ● “1”：小型，SNAT最大连接数10000 ● “2”：中型，SNAT最大连接数50000 ● “3”：大型，SNAT最大连接数200000 ● “4”：超大型，SNAT最大连接数1000000
router_id	String	路由器的ID。
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。
status	String	● 功能说明：公网NAT网关的状态。 ● 取值范围：资源状态说明。
admin_state_up	Boolean	● 解冻/冻结状态。 ● 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	公网NAT网关的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例
  ```
  GET https://{Endpoint}/v2.0/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
  ```
- 响应样例
  ```
  {
     "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"
     }
  }
  ```

状态码

请参考**状态码**。

9.1.1.4 更新公网 NAT 网关

功能介绍

更新公网NAT网关。

说明

admin_state_up = True & status = "ACTIVE"允许更新，支持更新名称、描述、规格

URI

PUT /v2.0/nat_gateways/{nat_gateway_id}

表 9-11 参数说明

参数	参数类型	是否必选	描述
nat_gateway_id	String	是	所属公网NAT网关的id。

请求消息

请求参数如**表9-12**所示。

表 9-12 请求参数

参数	是否必选	参数类型	描述
nat_gateway	是	Object	nat_gateway对象。详见**表9-13**。 必选字段：无，只有name，description和spec字段允许更新，更新操作时至少指定一项属性。

表 9-13 nat_gateway 字段说明

参数	是否必选	参数类型	描述
name	否	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。

参数	是否必选	参数类型	描述
description	否	String	公网NAT网关的描述，长度限制为255。
spec	否	String	公网NAT网关的规格。 取值为： ● “1”：小型，SNAT最大连接数10000 ● “2”：中型，SNAT最大连接数50000 ● “3”：大型，SNAT最大连接数200000 ● “4”：超大型，SNAT最大连接数1000000

响应消息

响应消息如表9-14所示。

表 9-14 响应参数

参数	参数类型	描述
nat_gateway	Object	nat_gateway对象。详见表9-15。

表 9-15 nat_gateway 字段说明

参数	参数类型	描述
id	String	公网NAT网关的id。
tenant_id	String	项目的ID。
name	String	公网NAT网关的名字，长度限制为64。 公网NAT网关的名字仅支持数字、字母、_（下划线）、-（中划线）、中文。
description	String	公网NAT网关的描述，长度限制为255。
spec	String	公网NAT网关的规格。 取值为： ● “1”：小型，SNAT最大连接数10000 ● “2”：中型，SNAT最大连接数50000 ● “3”：大型，SNAT最大连接数200000 ● “4”：超大型，SNAT最大连接数1000000

参数	参数类型	描述
router_id	String	路由器的ID。
internal_network_id	String	公网NAT网关下行口（DVR的下一跳）所属的network id。
status	String	• 功能说明：公网NAT网关的状态。 • 取值范围：资源状态说明。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	公网NAT网关的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例

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

- 响应样例

```
{
   "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"
   }
}
```

状态码

请参考状态码。

9.1.1.5 删除公网 NAT 网关

功能介绍

删除公网NAT网关。

URI

DELETE /v2.0/nat_gateways/{nat_gateway_id}

表 9-16 参数说明

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	所属公网NAT网关的id。

请求消息

无

响应消息

无

示例

- 请求样例
  DELETE https://{Endpoint}/v2.0/nat_gateways/a78fb3eb-1654-4710-8742-3fc49d5f04f8
- 响应样例
  无（STATUS CODE 204）

状态码

请参考**状态码**。

9.1.2 SNAT 规则

9.1.2.1 创建 SNAT 规则

功能介绍

创建SNAT规则。

说明

创建规则时，要求网关状态status = "ACTIVE"，要求网关管理员状态admin_state_up = True。

URI

POST /v2.0/snat_rules

请求消息

请求参数如**表9-17**所示。

表 9-17 请求参数

参数	是否必选	参数类型	描述
snat_rule	是	Object	snat_rule对象。详见表9-18。

表 9-18 snat_rule 字段说明

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	所属公网NAT网关的id。
network_id	否	String	规则使用的网络id。与cidr参数二选一。
cidr	否	String	cidr，可以是网段或者主机格式，与network_id参数二选一。 source_type=0时，cidr必须是vpc 子网网段的子集（不能相等）； source_type=1时，cidr必须指定专线侧网段。
source_type	否	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC）
floating_ip_id	是	String	功能说明：弹性公网IP的id，多个弹性公网IP的id使用逗号分隔。 取值范围：最大长度4096字节。 约束：弹性公网IP的id个数不能超过20个。

响应消息

响应参数如表9-19所示。

表 9-19 响应参数

参数	参数类型	描述
snat_rule	Object	snat_rule对象。详见表9-20。

表 9-20 snat_rule 字段说明

参数	参数类型	描述
id	String	SNAT规则的id。

参数	参数类型	描述
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
network_id	String	规则使用的网络id。
cidr	String	cidr，vpc子网网段的子集或专线侧网段。
source_type	Integer	0：VPC侧，可以指定network_id 或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC）
floating_ip_id	String	• 功能说明：弹性公网IP的id，多个弹性公网IP的id使用逗号分隔。 • 取值范围：最大长度4096字节。
floating_ip_address	String	• 功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。 • 取值范围：最大长度1024字节。
status	String	• 功能说明：SNAT规则的状态。 • 取值范围：**资源状态说明**。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	SNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例

  a. VPC 侧指定network_id

  ```
  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. VPC侧指定CIDR

  ```
  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. 专线侧 指定CIDR

```
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"
    }
}
```

- 响应样例

a. VPC 侧指定network_id的响应

```
{
    "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. VPC 侧指定CIDR的响应

```
{
    "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. 专线侧指定CIDR的响应

```
{
    "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",
        "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"
    }
}
```

状态码

请参考**状态码**。

9.1.2.2 查询 SNAT 规则列表

功能介绍

查询SNAT规则列表。

URI

GET /v2.0/snat_rules

说明

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。支持参数说明中所有非必选参数过滤，请参考请求样例。

表 9-21 参数说明

参数	是否必选	参数类型	描述
id	否	String	SNAT规则的id。
limit	否	Integer	每页返回的个数。
tenant_id	否	String	项目的ID。
nat_gateway_id	否	String	所属公网NAT网关的id。
network_id	否	String	规则使用的网络id。
cidr	否	String	cidr，vpc子网网段的子集或专线侧网段。
source_type	否	Integer	0：VPC侧，可以指定network_id或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC）
floating_ip_id	否	String	• 功能说明：弹性公网IP的id。 • 取值范围：长度限制为4096。
floating_ip_address	否	String	• 功能说明：弹性公网IP。 • 取值范围：长度限制为1024。
status	否	String	• 功能说明：SNAT规则的状态。 • 取值范围：资源状态说明。
admin_state_up	否	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结

参数	是否必选	参数类型	描述
created_at	否	String	SNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

请求消息

无

响应消息

响应参数如表9-22所示。

表 9-22 响应参数

参数	参数类型	描述
snat_rules	List(snat_rule)	snat_rule对象列表。详见表9-23。

表 9-23 snat_rule 字段说明

参数	参数类型	描述
id	String	SNAT规则的id。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
network_id	String	规则使用的网络id。
cidr	String	cidr，vpc子网网段的子集或专线侧网段。
source_type	Integer	0：VPC侧，可以指定network_id或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC）
floating_ip_id	String	• 功能说明：弹性公网IP的id，多个弹性公网IP的id使用逗号分隔。 • 取值范围：最大长度4096字节。 • 约束：弹性公网IP的id个数不能超过20个

参数	参数类型	描述
floating_ip_address	String	• 功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。 • 取值范围：最大长度1024字节。
status	String	• 功能说明：SNAT规则的状态。 • 取值范围：**资源状态说明**。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	SNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例
  ```
  GET https://{Endpoint}/v2.0/snat_rules?limit=10
  ```
- 响应样例
  ```
  {
     "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"
       }
     ]
  }
  ```

状态码

请参考**状态码**。

9.1.2.3 查询指定的 SNAT 规则详情

功能介绍

查询指定的SNAT规则详情。

URI

GET /v2.0/snat_rules/{snat_rule_id}

表 9-24 参数说明

参数	是否必选	参数类型	描述
snat_rule_id	是	String	所属SNAT规则的id。

请求消息

无

响应消息

响应参数如表9-25所示。

表 9-25 响应参数

参数	参数类型	描述
snat_rule	Object	snat_rule对象。详见表9-26。

表 9-26 snat_rule 字段说明

参数	参数类型	描述
id	String	SNAT规则的id。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
network_id	String	规则使用的网络id。
cidr	String	cidr，vpc子网网段的子集或专线侧网段。
source_type	Integer	0：VPC侧，可以指定network_id或者cidr 1：专线侧，只能指定cidr 不输入默认为0（VPC）

参数	参数类型	描述
floating_ip_id	String	• 功能说明：弹性公网IP的id，多个弹性公网IP的id使用逗号分隔。 • 取值范围：最大长度4096字节。 • 约束：弹性公网IP的id个数不能超过20个
floating_ip_address	String	• 功能说明：弹性公网IP，多个弹性公网IP使用逗号分隔。 • 取值范围：最大长度1024字节。
status	String	• 功能说明：SNAT规则的状态 • 取值范围：资源状态说明。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	SNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例

```
GET https://{Endpoint}/v2.0/snat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
```

- 响应样例

```
{
    "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"
    }
}
```

状态码

请参考状态码。

9.1.2.4 删除 SNAT 规则

功能介绍

删除SNAT规则。

URI

DELETE /v2.0/snat_rules/{snat_rule_id}

表 9-27 参数说明

参数	是否必选	参数类型	描述
snat_rule_id	是	String	所属SNAT规则的id。

请求消息

无

响应消息

无

示例

- 请求样例
  DELETE https://{Endpoint}/v2.0/snat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
- 响应样例
  无（STATUS CODE 204）

状态码

请参考**状态码**。

9.1.3 DNAT 规则

9.1.3.1 创建 DNAT 规则

功能介绍

创建DNAT规则

说明

创建规则时，要求网关状态status = "ACTIVE"，要求网关管理员状态admin_state_up = True。port_id和private_ip不能同时生效。对于all port类型的规则，要求internal_service_port = 0，external_service_port = 0，protocol = any.

URI

POST /v2.0/dnat_rules

请求消息

请求参数如表9-28所示。

表 9-28 请求参数

参数	是否必选	参数类型	描述
dnat_rule	是	Object	dnat_rule对象。详见表9-29。

表 9-29 dnat_rule 字段说明

参数	是否必选	参数类型	描述
nat_gateway_id	是	String	所属公网NAT网关的id。
port_id	否	String	虚拟机或者裸机的Port ID，与private_ip参数二选一。
private_ip	否	String	用户私有IP地址，例如专线连接的私有云地址，与port_id参数二选一。
internal_service_port	是	Integer	虚拟机或者裸机对外提供服务的协议端口号。 取值范围：0~65535。
floating_ip_id	是	String	弹性公网IP的id。
external_service_port	是	Integer	Floatingip对外提供服务的端口号。 取值范围：0~65535。
protocol	是	String	协议类型，目前支持TCP/UDP/ANY 对应协议号6/17/0
internal_service_port_range	否	String	虚拟机或者裸机对外提供服务的协议端口号范围。 • 功能说明：该端口范围与external_service_port_range按顺序实现1:1映射。 • 取值范围：1~65535。 • 约束：只能以’-’字符连接端口范围。

参数	是否必选	参数类型	描述
external_service_port_range	否	String	Floatingip对外提供服务的端口号范围。 • 功能说明：该端口范围与internal_service_port_range按顺序实现1:1映射。 • 取值范围：1~65535。 • 约束：只能以’-’字符连接端口范围。

响应消息

响应参数如表9-30所示。

表 9-30 响应参数

参数	参数类型	描述
dnat_rule	Object	dnat_rule对象。详见表9-31。

表 9-31 dnat_rule 字段说明

参数	参数类型	描述
id	String	DNAT规则的id。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
port_id	String	虚拟机或者裸机的Port ID，在VPC场景时使用此参数，与private_ip参数二选一。
private_ip	String	用户私有IP地址，例如专线连接的私有云地址， 在专线场景时使用此参数，与port_id参数二选一。
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。
floating_ip_id	String	弹性公网IP的id。
floating_ip_address	String	弹性公网IP的IP地址。
external_service_port	Integer	Floatingip对外提供服务的端口号。

参数	参数类型	描述
protocol	String	协议类型，目前支持TCP/UDP/ANY 对应协议号6/17/0
status	String	• 功能说明：DNAT规则的状态。 • 取值范围：资源状态说明。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	DNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss
internal_service_port_range	String	虚拟机或者裸机对外提供服务的协议端口号范围。 • 功能说明：该端口范围与external_service_port_range按顺序实现1:1映射。 • 取值范围：1~65535。 • 约束：只能以’-’字符连接端口范围。
external_service_port_range	String	Floatingip对外提供服务的端口号范围。 • 功能说明：该端口范围与internal_service_port_range按顺序实现1:1映射。 • 取值范围：1~65535。 • 约束：只能以’-’字符连接端口范围。

示例

- 请求样例

  a. 创建指定端口的规则

```
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. 创建all port类型的规则。

```
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
   }
}
```

c. 指定端口范围创建规则

```
POST https://{Endpoint}/v2.0/dnat_rules
 {
    "dnat_rule": {
      "floating_ip_id": "0cc38f0c-f26b-4556-b956-f5831061bb86",
      "nat_gateway_id": "dcb80bee-3e67-4282-8cc3-981431a63583",
      "private_ip": "172.16.1.197",
      "internal_service_port": 0,
      "internal_service_port_range": "55-66",
      "protocol": "udp",
      "external_service_port": 0,
      "external_service_port_range": "55-66",
      "description": "my dnat rule 01"
    }
 }
```

- 响应样例

a. 创建指定端口的规则的响应

```
{
   "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. 创建all port类型的规则的响应

```
{
   "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
   }
}
```

c. 指定端口范围创建规则的响应

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



```
        "floating_ip_id": "0cc38f0c-f26b-4556-b956-f5831061bb86",
        "status": "ACTIVE",
        "nat_gateway_id": "dcb80bee-3e67-4282-8cc3-981431a63583",
        "admin_state_up": true,
        "private_ip": "172.16.1.197",
        "internal_service_port": 0,
        "protocol": "udp",
        "tenant_id": "057ef081ad80d2732fcec011fdbc01c0",
        "created_at": "2020-09-21 11:46:11.474729",
        "id": "0de17f1a-686a-4484-9d8b-973889f8654c",
        "external_service_port": 0,
        "floating_ip_address": "10.185.74.219",
        "port_id": "",
        "internal_service_port_range":"55-66",
        "external_service_port_range":"55-66"
    }
}
```


状态码

请参考**状态码**。

9.1.3.2 查询 DNAT 规则列表

功能介绍

查询DNAT规则列表。

URI

GET /v2.0/dnat_rules

说明

可以在URI后面用‘?’和‘&’添加不同的查询条件组合。支持参数说明中所有非必选参数过滤，请参考请求样例。

表 9-32 参数说明

参数	参数类型	描述
id	String	DNAT规则的id。
limit	Integer	每页返回的个数。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
port_id	String	虚拟机或者裸机的Port ID。
private_ip	String	用户私有IP地址，例如专线连接的私有云地址。
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。
floating_ip_id	String	弹性公网IP的id。

参数	参数类型	描述
floating_ip_address	String	弹性公网的IP地址。
external_service_port	Integer	Floatingip对外提供服务的端口号。
protocol	String	协议类型，目前支持TCP/UDP/ANY 对应协议号6/17/0
status	String	• 功能说明：DNAT规则的状态。 • 取值范围：**资源状态说明**。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	DNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

请求消息

无

响应消息

响应参数如**表9-33**所示。

表 9-33 响应参数

参数	参数类型	描述
dnat_rules	Array(Object)	dnat_rule对象列表。详见**表9-34**。

表 9-34 dnat_rule 字段说明

参数	参数类型	描述
id	String	DNAT规则的id。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
port_id	String	虚拟机或者裸机的Port ID。

参数	参数类型	描述
private_ip	String	用户私有IP地址，例如专线连接的私有云地址。
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。
floating_ip_id	String	弹性公网IP的id。
floating_ip_address	String	弹性公网的IP地址。
external_service_port	Integer	Floatingip对外提供服务的端口号。
protocol	String	协议类型，目前支持TCP/UDP/ANY 对应协议号6/17/0
status	String	• 功能说明：DNAT规则的状态。 • 取值范围：**资源状态说明**。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	DNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例
  ```
  GET https://{Endpoint}/v2.0/dnat_rules
  ```
- 响应样例
  ```
  {
     "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
        }
    ]
}
```


状态码

请参考**状态码**。

9.1.3.3 查询指定的 DNAT 规则详情

功能介绍

查询指定的DNAT规则详情。

URI

GET /v2.0/dnat_rules/{dnat_rule_id}

表 9-35 参数说明

参数	参数类型	是否必选	描述
dnat_rule_id	String	是	所属DNAT规则的id。

请求消息

无

响应消息

响应参数如**表9-36**所示。

表 9-36 响应参数

参数	参数类型	描述
dnat_rule	Object	dnat_rule对象。详见**表9-37**。

表 9-37 dnat_rule 字段说明

参数	参数类型	描述
id	String	DNAT规则的id。
tenant_id	String	项目的ID。
nat_gateway_id	String	所属公网NAT网关的id。
port_id	String	虚拟机或者裸机的Port ID。
private_ip	String	用户私有IP地址，例如专线连接的私有云地址。
internal_service_port	Integer	虚拟机或者裸机对外提供服务的协议端口号。
floating_ip_id	String	弹性公网IP的id。
floating_ip_address	String	弹性公网的IP地址。
external_service_port	Integer	Floatingip对外提供服务的端口号。
protocol	String	协议类型，目前支持TCP/UDP/ANY 对应协议号6/17/0
status	String	• 功能说明：DNAT规则的状态。 • 取值范围：**资源状态说明**。
admin_state_up	Boolean	• 解冻/冻结状态。 • 取值范围： – “true”：解冻 – “false”：冻结
created_at	String	DNAT规则的创建时间戳，遵循UTC时间，保留小数点后6位，格式是yyyy-mm-dd hh:mm:ss

示例

- 请求样例
  ```
  GET https://{Endpoint}/v2.0/dnat_rules/5b95c675-69c2-4656-ba06-58ff72e1d338
  ```
- 响应样例
  ```
  {
    "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": ""
  }
}
```

状态码

请参考**状态码**。

9.1.3.4 删除 DNAT 规则

功能介绍

删除DNAT规则。

URI

DELETE /v2.0/dnat_rules/{dnat_rule_id}

表 9-38 参数说明

参数	是否必选	参数类型	描述
dnat_rule_id	是	String	所属DNAT规则的id。

请求消息

无

响应消息

无

示例

- 请求样例
  ```
  DELETE https://{Endpoint}/v2.0/dnat_rules/a78fb3eb-1654-4710-8742-3fc49d5f04f8
  ```
- 响应样例
  ```
  无（STATUS CODE 204）
  ```

状态码

请参考**状态码**。