

# SDN网络服务 使用手册

产品版本: v6.1.1  
发布日期: 2023-11-20

# 目录

1 版本说明 .....	1
1.1 版本说明书 .....	1
2 产品介绍 .....	3
2.1 什么是SDN网络服务 .....	3
2.2 使用场景 .....	7
2.3 基本概念 .....	8
2.4 产品获取 .....	9
2.5 权限说明 .....	10
2.6 使用限制 .....	13
2.7 与其他服务的关系 .....	14
3 快速入门 .....	15
3.1 操作指引 .....	15
3.2 前置条件准备 .....	18
3.3 创建二层基础网络 .....	19
3.4 配置公网IP池（可选） .....	22
3.5 配置三层互通网络（可选） .....	24
3.6 配置安全访问策略（可选） .....	26
3.7 配置多网卡访问（可选） .....	28
3.8 配置外网访问（可选） .....	30

---

3.9 查看网络拓扑 .....	32
4 用户指南 .....	33
4.1 网络 .....	33
4.2 虚拟网卡 .....	37
4.3 安全组 .....	41
4.4 路由器 .....	44
4.5 公网IP .....	48
4.6 网络拓扑 .....	51
5 最佳实践 .....	52
5.1 通过Keepalived与虚拟IP结合配置云主机网络 高可用 .....	52
5.2 配置多活网络方案 .....	56
5.3 配置路由器连接多个外网 .....	62
5.4 配置云内对等连接 .....	66
6 常见问题 .....	69
6.1 云主机IP地址未提供服务，如何排查解决 .....	69
7 API参考 .....	70
7.1 API简介 .....	70
7.2 调用方式 .....	72
7.3 网络 .....	78
7.4 子网 .....	93
7.5 虚拟网卡 .....	109

---

7.6 QoS策略 .....	127
7.7 路由器 .....	139
7.8 安全组 .....	161
7.9 公网IP .....	184
7.10 发布记录 .....	198

# 1 版本说明

## 1.1 版本说明书

### 版本信息

产品名称	产品版本	发布日期
SDN网络服务	V6.1.1	2022-05-31

### 更新说明

#### 新增功能

- 支持一个路由器连接到多个外部网络（互联网、办公网、专线等），不同外部网络可以指定网络节点实现分流和隔离。
- 一个Region内支持创建对等连接，不同路由器可以通过对等连接三层串联起来，管理员可以通过对等连接将不同租户网络联通。
- 支持一个路由器在路由模式下多活部署，用户可根据南北向业务流量需求平滑动态增删多活网络节点。
- 支持裸金属使用隧道Geneve网络，支持裸金属安全组，实现云主机、裸金属网络统一规划。
  - 裸金属网关节点支持多活部署，多个裸金属网关节点同时提供服务。
  - 裸金属网关节点支持高可用，任何裸金属网关故障后，其他裸金属网关可快速接管故障裸金属网关上的服务。
- 网络节点、裸金属网关节点支持DPDK模式，大幅提升南北向网络、裸金属网络带宽、PPS性能。
- 支持负载均衡云主机使用无状态安全组提升网络带宽、PPS性能。
- 支持网络监控告警能力。
  - 支持网络节点、裸金属网关节点网络监控。
  - 支持计算节点vSwitch监控。
  - 支持网络连通性监控。

## 优化功能

- 允许VLAN网络的VLAN范围不连续，方便用户管理业务VLAN。
- 优化公网IP实现机制，大幅提升公网IP的带宽、PPS性能。
- 提供同一子网内相同MAC的虚拟网卡检查机制，避免引起MAC冲突。

## 依赖说明

- 平台版本至少为V6.1.1。

## 2 产品介绍

### 2.1 什么是SDN网络服务

SDN网络服务旨在为云主机、容器、安全容器和裸金属等计算资源构建安全隔离的、自主配置和自主管理的虚拟网络环境，提升云上资源的安全性，简化网络的部署。客户可以按需配置子网、虚拟网卡和安全组等功能，并允许灵活搭配路由器和公网IP，支撑其业务部署。

### 产品优势

- **安全可靠**

构建的网络环境保证逻辑隔离，并提供完善的安全组策略，此外，还支持多网络可用区部署，提供高可靠的网络服务。

- **灵活配置**

通过软件定义网络的方式，实现子网、IP地址范围、安全组、路由器和公网IP等网络对象的按需配置和灵活定义。

- **高性能网络**

网络节点性能实现全面优化，即通过DPDK纵向提升性能，通过多活部署横向提升性能。

- **IPv4/IPv6双栈**

在同一套云平台中，通过统一架构，实现IPv4和IPv6双栈网络，降低使用复杂度和应用成本。

- **互联互通**

在安全隔离基础上，支持灵活配置网络之间互联互通，包括内外部网络之间和不同私有网络之间。

- **全栈网络**

支持云主机、容器、安全容器和裸金属共用网络规划，实现互联互通。

### 主要功能

## 网络

- **自定义虚拟网络**

支持在物理网络基础上构建二层隔离的虚拟网络，虚拟网络类型可以选择Geneve或VLAN，不同虚拟网络之间二层逻辑隔离。支持在虚拟网络中，创建IPv4、IPv6或者双栈的一个或多个子网。支持自定义子网的网段、网关、IP地址范围、DHCP等配置，按需对网络进行规划和管理。

- **自定义外部网络**

支持构建能通过任意路由器访问的外部网络，以满足虚拟网络访问公网、专线、办公网等的需求，外部网络类型可以选择VLAN或Flat。

- **自定义可用区**

支持为网络选择多个可用区，通过将节点划分入不同可用区的方式，实现节点之间的物理隔离。然后，通过在构建虚拟网络和路由器时选择独立的可用区，以满足业务隔离的需求。

- **多生产网**

支持构建多个物理网络，即在创建虚拟网络时，选择在指定的物理网络上构建，使在不同物理网络上构建的虚拟网络实现二层物理隔离。

- **实例互联互通**

支持在同一个虚拟网络中，云主机、容器、安全容器和裸金属等资源之间能够互联互通。

## 虚拟网卡

- **自定义虚拟网卡**

支持灵活配置虚拟网卡的IP地址、安全组等，并支持将虚拟网卡与云主机、容器、安全容器和裸金属等资源进行绑定与解绑。

- **IPv4/IPv6双栈**

支持灵活配置虚拟网卡的IP地址为IPv4、IPv6或IPv4/IPv6双栈类型，以满足客户多样化的组网需求。

- **绑定虚拟IP**



支持为多个虚拟网卡分配相同的虚拟IP地址，以满足高可用架构对于虚拟IP的需求，提供虚拟IP管理。

- **自定义带宽限制**

支持灵活配置虚拟网卡的QoS带宽，方便自定义虚拟网卡的带宽能力。

## 安全组

- **自定义安全组**

支持自行创建安全组，并绑定到相应的资源网卡上，使绑定相同安全组的网卡属于同一安全域。

- **自定义安全组规则**

支持灵活配置安全组的访问规则，即访问规则支持配置多种协议类型和多种匹配方式，以满足用户各类安全隔离规范的要求，而且安全组规则的更新能够自动应用到关联的资源网卡。

## 路由器

- **自定义互联子网**

支持通过绑定不同子网到同一路由器的方式，实现子网间互联互通的能力。

- **自定义路由规则**

支持根据业务需求灵活配置路由器路由规则，即通过设置目标网段和下一跳的方式，精细化管理网络流量的转发路径。

- **共享带宽**

支持在设置路由器网关时，使子网通过配置的SNAT，获得访问外部网络的能力。

- **自定义共享带宽限制**

支持灵活配置路由器QoS带宽，方便自定义路由器的带宽能力。

- **连接多外部网络**

支持将路由器同时连接到不同的外部网络，使其除连接公网外还可同时连接到办公网、专线等，方便构建混合云业务。

- **路由器多活**

支持自定义路由器ECMP等价路由连接到多个网络节点，以及水平灵活扩展多活网络节点，以满足高可用与高带宽的需求。

## 公网IP

- **绑定公网IP**

支持将创建的公网IP资源，根据需求绑定到云主机和裸金属主机等资源或路由器上，使资源或路由器获取与公网互相访问的能力。

- **自定义带宽限制**

支持灵活配置公网IP的QoS带宽，方便自定义公网IP的带宽能力。

## 拓扑展示

- **支持网络拓扑图形化展示**

支持直观查看云平台中的网络拓扑。拓扑图中包含已创建好的路由器、网络及各个子网下关联的云主机等资源信息。

## 2.2 使用场景

### 网络

- **业务系统独立网络部署**

对于安全性要求高的企业，通常需要按IT业务类型对内部网络进行隔离，可以通过使用SDN网络服务云产品，灵活按照内部业务类型，配置不同的网络，实现网络之间的逻辑隔离。

- **云主机、容器、裸金属统一网络**

SDN网络服务云产品支持云主机、容器、裸金属主机共享网络规划，统一进行管理，实现容器、云主机、裸金属主机之间的互联互通。

- **多业务隔离区与多生产网**

企业用户一朵云中根据监管或业务要求有业务网络物理隔离的需求，可以通过隔离区将业务网络物理隔离开，每个隔离区里有自己的网络、子网、路由器、公网IP等。

企业用户的某些业务系统对网络性能要求严格，对网络质量非常敏感，可以划分出单独的物理生产网，彻底解决存在的带宽争抢、资源被占用的问题。

- **公网、办公网、专线等多外部网络访问**

企业用户一般都有公网、办公网、专线网络等多种网络，对于云平台上的计算资源，会有多种网络访问需求，比如允许一些云主机只能访问公网、办公网、专线网络中的一种网络，或允许一些云主机同时可以访问三种网络的其中二种或三种网络。通过云平台提供的多外部网络搭配路由器功能可轻松实现按需的外部网络访问。

## 2.3 基本概念

- **网络**

与物理基础设施网络功能基本相同，云平台提供内部网络和外部网络两种网络类型。其中，内部网络主要用于为云主机和裸金属主机等资源提供二层隔离的私有网络环境。外部网络主要用于创建公网IP地址池。

- **虚拟网卡**

是绑定私有网络内资源的一种弹性网络接口，可在多个资源之间自由迁移。通过配置多个虚拟网卡，可以扩展资源原有的网络接入端口，使资源能够同时连接到多个网络链路。

- **安全组**

是一个逻辑分组，为具有相同安全保护需求并相互信任的资源提供相同的访问策略。

- **路由器**

通过提供三层路由功能，不仅可以使不同子网资源之间互联互通，还可以使内部资源与外部网络之间互相访问。

- **公网IP**

独立的IP地址资源。公网IP通过与资源绑定，可以使各资源独立对外提供服务，通过与路由器绑定，可以使不同子网之间或内部网络与外部网络之间相互访问。

- **网络可用区**

指在同一资源池内，独立提供DHCP服务以及三层网络服务的逻辑区域。

- **CIDR (Classless Inter-Domain Routing)**

即无类别域间路由，用于分配IP地址以及有效路由IP数据包并对IP地址进行归类。其是一种新的寻址方式，与传统的A类、B类和C类寻址模式相比，CIDR在IP地址分配方面更为高效。其采用斜线记法，表示为：IP地址/网络ID的位数。

## 2.4 产品获取

### 前提条件

在执行下述产品获取操作步骤前，请确保以下条件均已满足：

- 已成功卸载“基础网络服务云产品”。卸载云产品的具体操作说明，请参考“产品与服务管理”帮助中的相关内容。
- 如需获取正式版云产品，请提前将已获取的许可文件准备就绪。

### 操作步骤

1. 获取并安装SDN网络服务云产品。

在云平台的顶部导航栏中，依次选择[产品与服务]-[产品与服务管理]-[云产品]，进入“云产品”页面获取“SDN网络服务”云产品。具体的操作说明，请参考“产品与服务管理”帮助中“云产品”的相关内容。

2. 访问SDN网络服务。

在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]后，选择各子菜单，即可访问该服务的各项功能。

## 2.5 权限说明

本章节主要用于说明SDN网络服务各功能的用户权限范围。其中，√代表该类用户可对云平台内所有项目的操作对象执行此功能，**XX项目**代表该类用户仅支持对XX项目内的操作对象执行此功能，未标注代表该类用户无权限执行此功能。

功能		云管理员	部门管理员/项目管理员/普通用户
网络	信息展示	√	仅已加入项目
	创建网络		
	导出		
	编辑		
	创建/编辑/删除子网		
	编辑子网标签		
	编辑标签		
	删除网络		
虚拟网卡	信息展示	√	仅已加入项目
	创建虚拟网卡	√	
	导出	√	
	绑定到资源	仅Default/admin项目	
	从资源解绑	仅Default/admin项目	
	编辑	√	
	编辑安全组	仅Default/admin项目	

	功能	云管理员	部门管理员/项目管理员/普通用户
	绑定/解绑安全组	仅Default/admin项目	
	添加/删除安全组规则	仅Default/admin项目	
	管理虚拟IP	√	
	编辑标签	√	
	删除网卡	√	
安全组	信息展示	√	仅已加入项目
	创建安全组	√	
	添加/删除规则	仅Default/admin项目	
	编辑标签	√	
	删除安全组	√	
路由器	信息展示	√	仅已加入项目
	创建路由器		
	导出		
	编辑		
	连接/断开子网		
	添加/删除静态路由		
	更新带宽		
	设置/清除网关		
	编辑标签		
	删除路由器		

功能		云管理员	部门管理员/项目管理员/普通用户
公网IP	为项目申请IP	√	仅已加入项目
	导出		
	更新带宽		
	绑定到虚拟网卡		
	解除绑定		
	编辑标签		
	释放公网IP		
网络拓扑	查看网络拓扑	√	仅已加入项目
	开/关设备标签显示		
	开关资源折叠显示		



## 2.6 使用限制

- 路由器、外部网络暂不支持IPv6功能。而且，目前IPv6功能为技术预览版，尚在持续研发中，敬请期待。

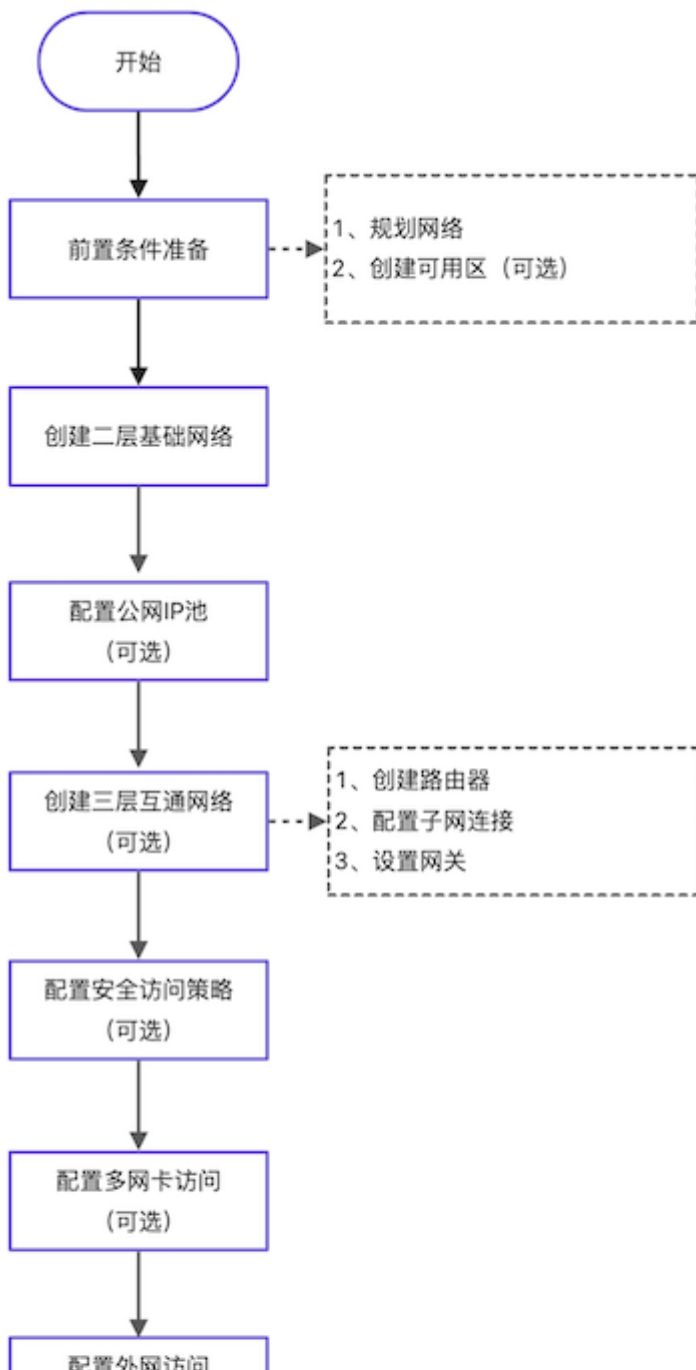
## 2.7 与其他服务的关系

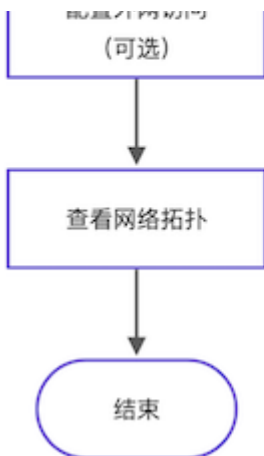
服务	说明
云监控服务	监控网络状态与告警。
身份与访问管理	提供鉴权服务。
标签服务	使用标签标识网络、虚拟网卡、安全组和路由器等网络对象，便于分类和搜索。

# 3 快速入门

## 3.1 操作指引

SDN网络服务云产品的主线使用流程及具体说明如下：





操作流程		描述
前置条件准备	规划网络	遵循隔离性、扩展性和连接性等因素，预先规划并设计网络。
	创建可用区（可选）	配置网络的可用区和主机集合，以提高其容灾性和隔离性。请根据客户实际业务需求酌情创建。如已有可用可用区或使用默认default-az时，可直接跳过本步骤。
创建二层基础网络		为云主机、容器、安全容器或裸金属等资源创建二层隔离的私有基础网络环境。
配置公网IP池（可选）		创建外部网络，为三层网络互通或外网访问需求配置所需的公网IP地址池。 请根据客户实际业务需求酌情创建。如无三层网络互通或外网访问需求或已有可用外部网络，可直接跳过本步骤。
配置三层互通网络（可选）	创建路由器	为云主机、容器、安全容器或裸金属等资源创建用于三层网络互连的路由器。
	配置子网连接	建立路由器与二层网络之间的连接，为各子网提供三层路由功能。
	设置网关	设置路由器网关，使其能够对外提供服务。
配置安全访问策略（可选）		为云主机或容器、安全容器等资源配置安全组策略，以提供相同的访问策略给具有相同安全保护需求并相互信任的资源。请根据客户实际业务需求酌情配置。如已有可用安全组或使用默认default时，可跳过本步骤。

操作流程	描述
配置多网卡访问（可选）	为云主机或裸金属等资源绑定多个虚拟网卡，扩展其原有网络接入端口，使资源能够同时连接到多个网络链路。 请根据客户实际业务需求酌情配置。当资源无多网卡访问需求时，可直接跳过本步骤。
配置外网访问（可选）	为云主机或裸金属等资源配置公网IP，使其能直接通过该公网IP对外提供服务。 请根据客户实际业务需求酌情配置。当资源无外网访问需求时，可直接跳过本步骤。
查看网络拓扑	直观明了地查看当前云平台的网络拓扑结构，确认各设备与资源之间的结构关系。

## 3.2 前置条件准备

在使用SDN网络服务前，请先完成以下准备工作。

### 规划网络

本操作用于根据客户的具体业务需求，对网络进行预规划和设计。

请根据客户的具体业务需求，遵循隔离性、扩展性和连接性等因素，规划并设计网络的设备类型、数量和网段等信息。

### 创建可用区（可选）

本操作用于为网络创建可用区和主机集合，以提高其容灾性和隔离性，请根据客户实际业务需求酌情创建。如有可用的可用区或使用默认default-az时，可跳过本步骤。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[计算]-[可用区与主机集合]，进入“可用区与主机集合”页面。
2. 单击 **创建主机集合** ，弹出“创建主机集合”对话框。输入集合名称和可用区名称后，单击 **创建** ，完成主机集合和可用区的创建。
3. 在“可用区与主机集合”页面中，选择上述新建的主机集合后，单击 **管理主机** ，添加节点至该主机集合和可用区中，并单击 **保存** ，完成操作。

## 3.3 创建二层基础网络

本操作用于在云平台中根据客户实际网络规划，为云主机、容器、安全容器或裸金属等资源创建二层隔离的私有基础网络环境。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，进入“网络”页面。
2. 单击 **创建网络** ，进入“创建网络”页面。
3. 配置参数后，单击 **创建网络** ，完成操作。其中，由于本操作创建的网络将用作云主机、容器、安全容器或裸金属等资源之间的网络通信，所以“网络类型”请选择“内部网络”。

参数		说明
基础配置	资源池	该网络所属的资源池。
	名称	该网络的名称。
	可用区	该网络所属的可用区。 当需要部署高可用网络时，请为其选择多个可用区，以便保证任何时间都有可用区的网络可使用，网络服务不中断。 Warning: 为确保云主机、容器、安全容器或裸金属等资源能够正常获取IP地址，请在资源创建时确保其可用区与该网络可用区之间的网络互通。
	网络类型	该网络的网络类型。该参数值可选内部网络或外部网络。 * 内部网络：用于云主机、容器、安全容器和裸金属等资源之间的网络通信。 * 外部网络：用作公网IP池，为云主机和路由器提供外网访问服务。
	可见性	该网络的可见范围。该参数值可选项目共享或全局共享。 * 项目共享：该网络仅可在所选项目内使用。 * 全局共享：该网络所有项目均可共享使用。

参数	说明	
网络模式	该网络的模式。 * 当网络类型为“内部网络”时，该参数值可选VLAN或Geneve。但是，当该内部网络需要配置三层互通时，该参数值仅能选择Geneve。 * 当网络类型为“外部网络”时，该参数值可选VLAN或Flat(No VLAN)。	
GENEVE VNI	该Geneve模式网络的网络标识符（VNI, Virtual Network Identifier）。 仅当网络模式选择“Geneve”时，才需设置此参数。	
物理网络	该网络所属的物理网络。 仅当网络模式选择“VLAN”或“Flat(No VLAN)”时，才需设置此参数。	
VLAN ID	该VLAN模式网络的ID。 仅当网络模式选择“VLAN”时，才需设置此参数。	
标签（可选）	该网络的标签。	
默认/新增IPv4子网	IPv4子网名称	该网络中，IPv4子网的名称。
	IPv4子网网段	该网络中，IPv4子网的网段。 该参数值可以手动输入，也可以单击输入框下方“推荐内网网段”的 <input type="button" value="选择"/> 自动指定。
	高级配置（可选）	该网络中，IPv4子网的网关地址和地址池范围等配置。 * 网关地址：用于设置该网络内云主机、容器、安全容器或裸金属等资源的缺省网关。 当该网络是外部网络时，请确保子网网关可达，否则该子网将无法正常使用。 * 地址池范围：用于精细化管理子网的IP地址。即网络可用的IP地址范围由此地址池指定。
新增IPv6子网 (仅内部网络支持)	IPv6子网名称	该网络中，IPv6子网的名称。



参数	说明
IPv6子网网段	该网络中，IPv6子网的网段。 目前仅提供Stateful-DHCP的地址分配方式，所以请以fd/FD/2/3为起始填写且前缀长度应该在64~125区间内。

## 3.4 配置公网IP池（可选）

本操作用于在云平台中创建用于分配公网IP地址的外部网络。当需要为云主机、容器、安全容器或裸金属等资源配置三层互通网络或配置外网访问时，可以使用系统默认创建的外部网络（public\_net）或已有的可用外部网络，也可以根据客户实际业务需求新建外部网络。如需新建外部网络，则必须执行本操作。否则，可直接跳过本步骤。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，进入“网络”页面。
2. 单击 **创建网络** ，进入“创建网络”页面。
3. 配置参数后，单击 **创建网络** ，完成操作。其中，由于本操作创建的网络将用作分配公网IP地址，所以“网络类型”请选择“外部网络”。

参数		说明
基础配置	资源池	该网络所属的资源池。
	名称	该网络的名称。
	可用区	该网络所属的可用区。 当需要部署高可用网络时，请为其选择多个可用区，以便保证任何时间都有可用区的网络可使用，网络服务不中断。
	网络类型	该网络的网络类型。该参数值可选内部网络或外部网络。 * 内部网络：用于云主机、容器、安全容器和裸金属等资源之间的网络通信。 * 外部网络：用作公网IP池，为云主机和路由器提供外网访问服务。
	可见性	该网络的可见范围。该参数值可选项目共享或全局共享。 * 项目共享：该网络仅可在所选项目内使用。 * 全局共享：该网络所有项目均可共享使用。

参数	说明	
网络模式	该网络的模式。 * 当网络类型为“内部网络”时，该参数值可选VLAN或Geneve。但是，当该内部网络需要配置三层互通时，该参数值仅能选择Geneve。 * 当网络类型为“外部网络”时，该参数值可选VLAN或Flat(No VLAN)。	
GENEVE VNI	该Geneve模式网络的网络标识符（VNI，Virtual Network Identifier）。 仅当网络模式选择“Geneve”时，才需设置此参数。	
物理网络	该网络所属的物理网络。 仅当网络模式选择“VLAN”或“Flat(No VLAN)”时，才需设置此参数。	
VLAN ID	该VLAN模式网络的ID。 仅当网络模式选择“VLAN”时，才需设置此参数。	
标签（可选）	该网络的标签。	
默认/新增IPv4子网	IPv4子网名称	该网络中，IPv4子网的名称。
	IPv4子网网段	该网络中，IPv4子网的网段。 该参数值可以手动输入，也可以单击输入框下方“推荐内网网段”的 <input type="checkbox"/> 选择 <input type="checkbox"/> 自动指定。
	高级配置（可选）	该网络中，IPv4子网的网关地址和地址池范围等配置。 * 网关地址：用于设置该外部网络的缺省网关，请确保子网网关可达，否则该子网将无法正常使用。 * 地址池范围：用于精细化管理子网的IP地址。即网络可用的IP地址范围由此地址池指定。

## 3.5 配置三层互通网络（可选）

本操作用于在云平台中根据客户实际网络规划，为云主机、容器、安全容器或裸金属等资源创建三层互通的网络环境，以便不同子网资源之间的互相联通，或内网资源与外部网络之间的互相访问。当资源无需跨子网访问和无需访问外部网络时，可直接跳过本步骤。

### 创建路由器

本操作用于在云平台中创建路由器，以作为不同子网之间或内网与外网之间的三层网络互连设备。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[路由器]，进入“路由器”页面。
2. 单击 **创建路由器** ，弹出“创建路由器”对话框。
3. 配置参数后，单击 **创建** ，完成操作。

参数	说明
名称	该路由器的名称。
可用区	该路由器的可用区。 当部署高可用网络时，请为其选择与网络高可用相一致的多个可用区，以保证任何时间都有可用区的网络可使用，网络服务不中断。

### 配置子网连接

本操作用于建立路由器与二层网络之间的连接，为各子网提供三层路由功能，以满足业务访问需要。

1. 在“路由器”页面中，单击上述路由器名称，进入其详情页面。
2. 在[路由器连接]页签中，单击 **连接子网** ，弹出“连接子网”对话框。
3. 配置参数后，单击 **连接** ，完成操作。

参数	说明
----	----

参数	说明
路由器名称(不可编辑)	该路由器的名称。
子网	该路由器所连接的子网的名称。 请选择上述创建二层基础网络操作中新建网络的子网。
子网IP	该路由器所连接的子网的IP地址。 当所连接的子网已设置网关时，此IP地址只能选择子网的网关地址。

## 设置网关

本操作用于为路由器设置网关，使其能够使用此IP地址对外提供服务。

- 在上述路由器的详情页面中，单击 **更多操作** - **设置网关**，弹出“设置路由器网关”对话框。
- 配置参数后，单击 **设置**，完成操作。

参数	说明
外部网络	该路由器网关地址的所属外部网络。
分配外部IP	该路由器网关地址的分配方式。该参数值可选自动分配或手动选择。 当选择“手动选择”时，请手动选择子网和外部IP地址信息。
QoS	该路由器网关的QoS（Quality of Service，服务质量）功能是否激活。 当选择激活QoS时，请设置该路由器网关的带宽限制。
开启SNAT	该路由器网关是否开启SNAT（Source Network Address Translation，源地址转换）。 * 当开启SNAT时，云主机、容器、安全容器或裸金属等资源在进行三层通信时，源IP地址将修改为该路由器网关地址。 * 当未开启SNAT时，云主机、容器、安全容器或裸金属等资源在进行三层通信时，将通过路由转发的方式实现。

## 3.6 配置安全访问策略（可选）

本操作用于在云平台中根据客户实际网络规划，为云主机或容器、安全容器等资源创建安全组并定义各种访问规则，用于为具有相同安全保护需求并相互信任的资源提供相同的访问策略。如已有可用安全组或使用默认 default 时，可跳过本步骤。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[安全组]，进入“安全组”页面。
2. 在“安全组”页面中，单击 **创建安全组**，弹出“创建安全组”对话框。配置参数后，单击 **创建**，完成安全组创建。

参数	说明
名称	该安全组的名称。
描述（可选）	该安全组的描述信息。

3. 在“安全组”页面中，单击上述安全组所在行的“展开”图标，展开其“规则”区域框。在“规则”区域框中，单击 **添加规则**，弹出“添加规则”对话框。配置参数后，单击 **保存**，完成操作。

说明：

当指定的安全组规则需要删除时，请直接单击该规则所在行的 **删除**，然后，在弹出的“删除安全组规则”提示框中，单击 **删除** 即可。

参数	说明
定制规则（可选）	预先定制的常用安全组规则，包括定制TCP规则、定制UDP规则、定制GRE规则、定制VRRP规则和ALL。
其他协议（可选）	该安全组规则的IP协议。该参数值可选ALL TCP、ALL UDP、ALL ICMP、DNS、HTTP、HTTPS、IMAP、IMAPS、LDAP、MS SQL、MYSQL、POP3、POP3 S、RDP、SMTP、SMTPS或SSH。

参数	说明
方向	该安全组规则的作用方向。该参数值可选入口或出口。 仅当选择定制规则或选择ALL TCP、ALL UDP、ALL ICMP协议时，才需设置此参数。
打开端口	该安全组规则的作用范围。该参数值可选端口或端口范围。 * 当选择“端口”时，请输入具体需要打开的端口。该安全组规则将仅作用于该端口。 * 当选择“端口范围”时，请输入需要打开的端口范围（即起始端口和终止端口）。该安全组规则将作用于该范围的所有端口。 仅当定制规则选择“定制TCP规则”和“定制UDP规则”时，才需设置此参数。
远端	该安全组规则的作用对象。该参数值可选CIDR或当前安全组。 * 当选择“CIDR”时，请输入具体地址。该安全组规则将作用于该地址段。 * 当选择“当前安全组”时，该安全组规则将作用于所有选择当前安全组的资源，而且各资源之间可以通过指定的协议互相访问。当安全组发生更新时，规则将同步更新至所有选择此安全组的资源。

## 3.7 配置多网卡访问（可选）

本操作通过为云主机或裸金属等资源绑定多个虚拟网卡，扩展其原有网络接入端口，使资源能够同时连接到多个网络链路。请根据客户实际业务需求酌情配置。当资源无多网卡访问需求时，可直接跳过本步骤。

### 1. 创建虚拟网卡。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[虚拟网卡]，进入“虚拟网卡”页面。
2. 单击 **创建虚拟网卡** ，弹出“创建虚拟网卡”对话框。
3. 配置参数后，单击 **创建** ，完成操作。

参数	说明
网卡名称	该虚拟网卡的名称。
项目	该虚拟网卡的所属项目。
类型	该虚拟网卡的类型。该参数值可选IPv4、IPv6或网络双栈（即同时设置IPv4和IPv6网卡）。 请在选择类型后，设置对应的子网和IP地址。其中，IP地址不能与所选的二层基础网络的子网网关相同。
MAC地址	该虚拟网卡的Mac地址。该参数值可选自动分配或手动输入。 当选择“手动选择”时，请手动选择Mac地址信息。
安全组	该虚拟网卡绑定的安全组。
QoS	该虚拟网卡的QoS（Quality of Service，服务质量）功能是否激活。 当选择激活QoS时，请设置该虚拟网卡的带宽限制。

### 2. 绑定虚拟网卡到资源。

1. 在“虚拟网卡”页面中，勾选上述虚拟网卡后，单击 **绑定到资源** ，弹出“绑定资源”对话框
2. 配置参数后，单击 **绑定** ，完成操作。



---

参数	说明
网卡名称（不可编辑）	该虚拟网卡的名称。
资源	该虚拟网卡将绑定的具体资源。

## 3.8 配置外网访问（可选）

本操作通过为云主机或裸金属等资源配置公网IP，使其能直接通过该公网IP对外提供服务。当资源无外网访问需求时，可直接跳过本步骤。

### 1. 为项目申请IP。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[公网IP]，进入“公网IP”页面。
2. 单击 **为项目申请IP** ，弹出“为项目申请IP”对话框。
3. 配置参数后，单击 **分配** ，完成操作。

参数	说明
项目	本次申请公网IP地址的项目。
资源池	该公网IP地址所属的资源池。
公网IP	本次申请的公网IP地址。
带宽	该公网IP地址的带宽。

### 2. 绑定到虚拟网卡。

1. 在“公网IP”页面中，勾选上述公网IP后，单击 **更多** - **绑定到虚拟网卡** ，弹出“绑定到虚拟网卡”对话框。
2. 配置参数后，单击 **绑定** ，完成操作。

说明：

- 公网IP只能绑定给类型为“IPv4”或“网络双栈”的虚拟网卡。
- 当虚拟网卡所处子网未连接到路由器，或者连接的路由器未设置网关时，其将不会显示在此对话框的虚拟网卡下拉框中。

参数	说明
公网IP (不可编辑)	该公网IP的地址。
部门 (不可编辑)	该公网IP的所属部门。
项目 (不可编辑)	该公网IP的所属项目。
虚拟网卡是否绑定到资源	该公网IP是否绑定到资源。该参数值可选是或否。在本操作中, 请选择“是”。
虚拟网卡	该公网IP所绑定的虚拟网卡。

## 3.9 查看网络拓扑

本操作用于直观明了地查看当前云平台的网络拓扑结构，以确认各设备与资源之间的结构关系。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络拓扑]，进入“网络拓扑”页面。
2. 在“网络拓扑”页面中，查看当前云平台的网络拓扑结构图，确认各设备与资源之间的结构关系。

# 4 用户指南

## 4.1 网络

本章节主要介绍在网络页面中，针对网络的一系列运维管理操作，如：查看详情、导出、编辑、编辑标签和删除网络等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，即可进入“网络”页面。

### 查看详情

在“网络”页面中，单击网络名称，可进入其详情页面。在详情页面中，可查看该网络的基本信息、标签和子网。

在指定网络的详情页面中，还支持对网络执行编辑名称、创建子网、查看子网详情、编辑子网、编辑子网标签和删除子网等操作。具体操作说明如下：

### 编辑名称

- 在“基本信息”区域框中，单击名称后的“编辑”图标，弹出“编辑”对话框。
- 配置参数后，单击 **保存**，完成操作。

### 创建子网

- 在“子网”区域框中，单击 **创建子网**，进入“创建子网”页面。
- 配置参数后，单击 **创建子网**，完成操作。

说明：

- 在同一网络中创建多个子网时，各子网之间的网段不能重复。
- 在同一网络中，最多只能创建一个IPv6子网。

参数	说明
----	----

参数		说明
类型		该子网的类型。该参数值可选IPv4或IPv6。
网络（不可编辑）		该子网的所属网络。
IPv4子网名称		该IPv4子网的名称。
IPv4子网网段		该IPv4子网的网段。 该参数值可以手动输入，也可以单击输入框下方“推荐内网网段”的 <b>选择</b> 自动指定。
高级配置（可选）	网关地址	该IPv4子网的网关信息。 如需设置网关，请先勾选 <b>设置网关</b> ，再输入网关地址。 当该子网所属网络是外部网络时，请确保子网网关可达，否则该子网将无法正常使用。
	DHCP服务	该IPv4子网是否开启DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）服务。 云平台将默认保留1个IP地址，用于DHCP服务。
	DNS服务器地址（可选）	该IPv4子网的DNS服务器地址。
	地址池范围（可选）	该IPv4子网可用的IP地址范围，用于精细化管理子网的IP地址。
	主机路由（可选）	该IPv4子网的主机路由。 当添加主机路由后，云平台将为该IPv4子网内的所有云主机、容器、安全容器或裸金属等资源自动添加对应的静态路由。
IPv6子网名称		该IPv6子网的名称。
IPv6子网网段		该IPv6子网的网段。 目前仅提供Stateful-DHCP的地址分配方式，所以请以fd/FD/2/3为起始填写且前缀长度应该在64~125区间内。

## 查看子网详情

在“子网”区域框中，单击子网名称，可进入其详情页面。在详情页面中，可查看该子网的基本信息、高级设置和标签。

在指定子网的详情页面中，还支持对子网执行编辑名称操作。具体操作说明如下：

1. 在“基本信息”区域框中，单击名称后的“编辑”图标，弹出“编辑”对话框。
2. 配置参数后，单击 **保存**，完成操作。

## 编辑子网

1. 在“子网”区域框中，勾选待操作子网后，单击 **更多** - **编辑**，进入“编辑子网”页面。
2. 配置参数后，单击 **保存**，完成操作。

说明：

在编辑子网后，该更新配置仅对后续新建的资源生效，不会影响已有资源。

## 编辑子网标签

1. 在“子网”区域框中，勾选待操作子网后，单击 **更多** - **编辑标签**，弹出“编辑标签”对话框。
2. 配置参数后，单击 **保存**，完成操作。

说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 删除子网

1. 在“子网”区域框中，勾选待操作子网后，单击 **删除子网**，弹出“删除子网”提示框。
2. 单击 **删除**，完成操作。

## 导出

在“网络”页面中，单击列表上方的 **导出** ，可将当前云平台所有网络的信息导出到CSV文件中。

## 编辑

1. 在“网络”页面中，勾选待操作网络后，单击 **编辑** ，弹出“编辑”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 编辑标签

1. 在“网络”页面中，勾选待操作网络后，单击 **编辑标签** ，弹出“编辑标签”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

### 说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 删除网络

1. 在“网络”页面中，勾选待操作网络后，单击 **删除网络** ，弹出“删除网络”提示框。
2. 单击 **删除** ，完成操作。

### 警告：

- 删除后的网络无法恢复，请谨慎操作。
- 在删除网络前，请先断开该网络与各资源或路由器等设备的连接。



## 4.2 虚拟网卡

本章节主要介绍在虚拟网卡页面中，针对虚拟网卡的一系列运维管理操作，如：创建虚拟网卡、导出、查看详情、绑定到资源、从资源解绑、编辑、编辑安全组、管理虚拟IP、编辑标签和删除网卡等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[虚拟网卡]，即可进入“虚拟网卡”页面。

### 创建虚拟网卡

1. 在“虚拟网卡”页面中，单击 **创建虚拟网卡** ，弹出“创建虚拟网卡”对话框。
2. 配置参数后，单击 **创建** ，完成操作。

参数	说明
网卡名称	该虚拟网卡的名称。
项目	该虚拟网卡的所属项目。
类型	该虚拟网卡的类型。该参数值可选IPv4、IPv6或网络双栈（即同时设置IPv4和IPv6网卡）。 请在选择类型后，设置对应的子网和IP地址。其中，IP地址不能与所选的二层基础网络的子网网关相同。
MAC地址	该虚拟网卡的Mac地址。该参数值可选自动分配或手动输入。 当选择“手动选择”时，请手动选择Mac地址信息。
安全组	该虚拟网卡绑定的安全组。
QoS	该虚拟网卡的QoS（Quality of Service，服务质量）功能是否激活。 当选择激活QoS时，请设置该虚拟网卡的带宽限制。

### 查看详情

在“虚拟网卡”页面中，单击虚拟网卡名称，可进入其详情页面。在详情页面中，可查看该虚拟网卡的基本信息、标签和安全组信息。

在指定虚拟网卡的详情页面中，还支持对虚拟网卡执行编辑名称、绑定安全组、添加安全组规则、删除安全组规则和解绑安全组操作。具体操作说明如下：

## 编辑名称

1. 在“基本信息”区域框中，单击名称后的“编辑”图标，弹出“编辑名称”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 绑定安全组

1. 在“安全组”区域框中，单击 **绑定安全组** ，弹出“绑定安全组”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

说明：

该绑定安全组仅应用于此虚拟网卡，不会应用到所在云主机的其他虚拟网卡。

## 添加安全组规则

1. 在“安全组”区域框中，单击待操作安全组所在行的“展开”图标，展开规则区域框后，单击 **添加规则** ，弹出“添加规则”对话框。
2. 配置参数后，单击 **保存** ，完成操作。其中，各参数的具体说明，请参考 [添加安全组规则](#)。

## 删除安全组规则

1. 在“安全组”区域框中，单击待操作安全组所在行的“展开”图标，展开规则区域框后，单击待操作规则所在行的 **删除** ，弹出“删除安全组规则”提示框。
2. 单击 **删除** ，完成操作。

## 解绑安全组

1. 在“安全组”区域框中，勾选待操作安全组后，单击 **解绑安全组** ，弹出“解绑安全组”提示框。
2. 单击 **解绑安全组** ，完成操作。

## 导出

在“虚拟网卡”页面中，单击列表上方的 **导出** ，可将当前云平台所有虚拟网卡的信息导出到CSV文件中。

## 绑定到资源

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **绑定到资源** ，弹出“绑定资源”对话框。
2. 配置参数后，单击 **绑定** ，完成操作。

## 从资源解绑

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **从资源解绑** ，弹出“从资源解绑网卡”提示框。
2. 单击 **解绑** ，完成操作。

说明：

- 停止状态的云主机资源无法将虚拟网卡从资源解绑。
- 在将虚拟网卡从资源解绑时，请先删除该虚拟网卡上添加的所有虚拟IP。
- 当云主机移入回收站后，其绑定的虚拟网卡不能自动解绑，同时也不允许手动解绑。如需释放该虚拟网卡资源，请在将云主机移入回收站前进行手动解绑，或等待云主机彻底删除后自动释放。

## 编辑

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **更多** - **编辑** ，弹出“编辑网卡”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 编辑安全组

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **更多** - **编辑安全组** ，弹出“编辑安全组”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 管理虚拟IP

本操作用于为虚拟网卡配置虚拟IP，通常可以将同一个虚拟IP与多个资源的虚拟网卡进行绑定，作为用户自定义高可用服务的可漂移网络入口，保障业务应用的高可用性。

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **更多** - **管理虚拟IP** ，弹出“管理虚拟IP”对话框。
2. 单击 **添加虚拟IP** ，弹出“添加虚拟IP”对话框。配置参数后，单击 **保存** ，完成操作。

说明：

- 当指定虚拟网卡需要添加多个虚拟IP时，单击 **添加虚拟IP** ，逐个添加即可。
- 当指定虚拟网卡需要删除虚拟IP时，单击待操作虚拟IP后的“删除”图标，即可进行删除。
- 请合理规划虚拟IP与云平台中虚拟网卡的实际私网IP，避免冲突。此外，为了避免冲突，建议在成功添加一个虚拟IP后，在云平台的[虚拟网卡]页面，新建一个虚拟网卡，将其IP地址指定为当前配置的虚拟IP地址，且不能将该虚拟网卡挂载至资源上（建议在新建该虚拟网卡时名称中写入不能挂载的提示）。

## 编辑标签

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **更多** - **编辑标签** ，弹出“编辑标签”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 删除网卡

1. 在“虚拟网卡”页面中，勾选待操作虚拟网卡后，单击 **删除网卡** ，弹出“删除虚拟网卡”提示框。
2. 单击 **删除** ，完成操作。

## 4.3 安全组

本章节主要介绍在安全组页面中，针对安全组的一系列运维管理操作，如：创建安全组、查看详情、编辑标签和删除安全组等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[安全组]，即可进入“安全组”页面。

### 创建安全组

1. 在“安全组”页面中，单击 **创建安全组** ，弹出“创建安全组”对话框。
2. 配置参数后，单击 **创建** ，完成操作。

参数	说明
名称	该安全组的名称。
描述（可选）	该安全组的描述信息。

### 查看详情

在“安全组”页面中，单击待操作安全组所在行的“展开”图标，可展开其“规则”区域框。在该区域框中，可查看该安全组的所有规则信息。

在指定安全组的“规则”区域框中，还支持对安全组执行添加规则和删除规则操作。具体操作说明如下：

### 添加规则

1. 在“规则”区域框中，单击 **添加规则** ，弹出“添加规则”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

参数	说明
----	----

参数	说明
定制规则（可选）	预先定制的常用安全组规则，包括定制TCP规则、定制UDP规则、定制GRE规则、定制VRRP规则和ALL。
其他协议（可选）	该安全组规则的IP协议。该参数值可选ALL TCP、ALL UDP、ALL ICMP、DNS、HTTP、HTTPS、IMAP、IMAPS、LDAP、MS SQL、MYSQL、POP3、POP3 S、RDP、SMTP、SMTPS或SSH。
方向	该安全组规则的作用方向。该参数值可选入口或出口。 仅当选择定制规则或选择ALL TCP、ALL UDP、ALL ICMP协议时，才需设置此参数。
打开端口	该安全组规则的作用范围。该参数值可选端口或端口范围。 * 当选择“端口”时，请输入具体需要打开的端口。该安全组规则将仅作用于该端口。 * 当选择“端口范围”时，请输入需要打开的端口范围（即起始端口和终止端口）。该安全组规则将作用于该范围的所有端口。 仅当定制规则选择“定制TCP规则”和“定制UDP规则”时，才需设置此参数。
远端	该安全组规则的作用对象。该参数值可选CIDR或当前安全组。 * 当选择“CIDR”时，请输入具体地址。该安全组规则将作用于该地址段。 * 当选择“当前安全组”时，该安全组规则将作用于所有选择当前安全组的资源，而且各资源之间可以通过指定的协议互相访问。当安全组发生更新时，规则将同步更新至所有选择此安全组的资源。

## 删除规则

1. 在“规则”区域框中，单击待操作规则所在行的 **删除**，弹出“删除安全组规则”提示框。
2. 单击 **删除**，完成操作。

## 编辑标签

1. 在“安全组”页面中，勾选待操作安全组后，单击 **编辑标签**，弹出“编辑标签”对话框。
2. 配置参数后，单击 **保存**，完成操作。

说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 删除安全组

1. 在“安全组”页面中，勾选待操作安全组后，单击 **删除安全组** ，弹出“删除安全组”提示框。
2. 单击 **删除** ，完成操作。

## 4.4 路由器

本章节主要介绍在路由器页面中，针对路由器的一系列运维管理操作，如：创建路由器、查看详情、导出、编辑、更新带宽、设置网关、清除网关、编辑标签和删除路由器等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[路由器]，即可进入“路由器”页面。

### 创建路由器

1. 在“路由器”页面中，单击 **创建路由器** ，弹出“创建路由器”对话框。
2. 配置参数后，单击 **创建** ，完成操作。

参数	说明
名称	该路由器的名称。
可用区	该路由器的可用区。 当部署高可用网络时，请为其选择与网络高可用相一致的多个可用区，以保证任何时间都有可用区的网络可使用，网络服务不中断。

### 查看详情

在“路由器”页面中，单击路由器名称，可进入其详情页面。在详情页面中，可查看该路由器的基本信息、标签、路由器连接和静态路由信息。

在指定路由器的详情页面中，还支持对路由器执行编辑名称、连接子网、断开子网连接、添加静态路由和删除静态路由操作。具体操作说明如下：

#### 编辑名称

1. 在“基本信息”区域框中，单击名称后的“编辑”图标，弹出“编辑”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

#### 连接子网



1. 在[路由器连接]页签中，单击 **连接子网** ，弹出“连接子网”对话框。
2. 配置参数后，单击 **连接** ，完成操作。

参数	说明
路由器名称(不可编辑)	该路由器的名称。
子网	该路由器所连接的子网的名称。 请选择上述创建二层基础网络操作中新建网络的子网。
子网IP	该路由器所连接的子网的IP地址。 当所连接的子网已设置网关时，此IP地址只能选择子网的网关地址。

## 断开子网连接

1. 在[路由器连接]页签中，勾选待操作子网后，单击 **断开连接** ，弹出“断开连接”提示框。
2. 单击 **断开连接** ，完成操作。

## 添加静态路由

本操作用于为路由器手动添加路由信息，使匹配指定目的CIDR的数据包按照固定的下一跳路径进行传输。

1. 在[静态路由]页签中，单击 **添加静态路由** ，弹出“添加静态路由”对话框。
2. 配置参数后，单击 **添加** ，完成操作。

参数	说明
目的CIDR	该静态路由的目的CIDR地址段。
下一跳	该静态路由的下一跳IP地址。

## 删除静态路由

1. 在[静态路由]页签中，勾选待操作静态路由后，单击 **删除静态路由** ，弹出“添加静态路由”提示框。
2. 单击 **删除** ，完成操作。

## 导出

在“路由器”页面中，单击列表上方的 **导出** ，可将当前云平台所有路由器的信息导出到CSV文件中。

## 编辑

1. 在“路由器”页面中，勾选待操作路由器后，单击 **编辑** ，弹出“编辑路由器”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 更新带宽

1. 在“路由器”页面中，勾选待操作路由器后，单击 **更新带宽** ，弹出“更新带宽”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

## 设置网关

1. 在“路由器”页面中，勾选待操作路由器后，单击 **更多** - **设置网关** ，弹出“设置路由器网关”对话框。
2. 配置参数后，单击 **设置** ，完成操作。

参数	说明
外部网络	该路由器网关地址的所属外部网络。
分配外部IP	该路由器网关地址的分配方式。该参数值可选自动分配或手动选择。 当选择“手动选择”时，请手动选择子网和外部IP地址信息。
QoS	该路由器网关的QoS（Quality of Service，服务质量）功能是否激活。 当选择激活QoS时，请设置该路由器网关的带宽限制。

参数	说明
开启SNAT	<p>该路由器网关是否开启SNAT（Source Network Address Translation，源地址转换）。</p> <p>* 当开启SNAT时，云主机、容器、安全容器或裸金属等资源在进行三层通信时，源IP地址将修改为该路由器网关地址。</p> <p>* 当未开启SNAT时，云主机、容器、安全容器或裸金属等资源在进行三层通信时，将通过路由转发的方式实现。</p>

## 清除网关

1. 在“路由器”页面中，勾选待操作路由器后，单击 **更多** - **清除网关**，弹出“清除路由器网关”提示框。
2. 单击 **清除**，完成操作。

## 编辑标签

1. 在“路由器”页面中，勾选待操作路由器后，单击 **更多** - **编辑标签**，弹出“编辑标签”对话框。
2. 配置参数后，单击 **确认**，完成操作。

说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 删除路由器

1. 在“路由器”页面中，勾选待操作路由器后，单击 **删除路由器**，弹出“删除路由器”提示框。
2. 单击 **删除**，完成操作。

## 4.5 公网IP

本章节主要介绍在公网IP页面中，针对公网IP的一系列运维管理操作，如：为项目申请IP、导出、更新带宽、绑定到虚拟网卡、解除绑定、编辑标签和释放公网IP等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[公网IP]，即可进入“公网IP”页面。

### 为项目申请IP

1. 在“公网IP”页面中，单击 **为项目申请IP** ，弹出“为项目申请IP”对话框。
2. 配置参数后，单击 **分配** ，完成操作。

参数	说明
项目	本次申请公网IP地址的项目。
资源池	该公网IP地址所属的资源池。
公网IP	本次申请的公网IP地址。
带宽	该公网IP地址的带宽。

### 导出

在“公网IP”页面中，单击列表上方的 **导出** ，可将当前云平台所有公网IP的信息导出到CSV文件中。

### 更新带宽

1. 在“公网IP”页面中，勾选待操作公网IP后，单击 **更多** - **更新带宽** ，弹出“更新带宽”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

### 绑定到虚拟网卡

1. 在“公网IP”页面中，勾选待操作公网IP后，单击 **更多** - **绑定到虚拟网卡** ，弹出“绑定到虚拟网卡”对话框。
2. 配置参数后，单击 **绑定** ，完成操作。

说明：

- 公网IP只能绑定给类型为“IPv4”或“网络双栈”的虚拟网卡。
- 当虚拟网卡被删除后，其挂载的公网IP将自动解绑并重新回到可用状态。
- 当虚拟网卡所处子网未连接到路由器，或者连接的路由器未设置网关时，其将不会显示在此对话框的虚拟网卡下拉框中。

参数	说明
公网IP（不可编辑）	该公网IP的地址。
部门（不可编辑）	该公网IP的所属部门。
项目（不可编辑）	该公网IP的所属项目。
虚拟网卡是否绑定到资源	该公网IP是否绑定到资源。该参数值可选是或否。
虚拟网卡	该公网IP所绑定的虚拟网卡。

## 解除绑定

1. 在“公网IP”页面中，勾选待操作公网IP后，单击 **更多** - **解除绑定** ，弹出“解除绑定”提示框。
2. 单击 **解除** ，完成操作。

## 编辑标签

1. 在“公网IP”页面中，勾选待操作网络后，单击 **更多** - **编辑标签** ，弹出“编辑标签”对话框。
2. 配置参数后，单击 **保存** ，完成操作。

说明：

当需要添加标签时，除支持选择已创建的标签之外，还支持直接输入需要新建的标签。新建标签将在关闭对话框后，同步添加至云平台的标签服务中。

## 释放公网IP

1. 在“公网IP”页面中，勾选待操作网络后，单击 **释放公网IP** ，弹出“释放公网IP”提示框。
2. 单击 **释放** ，完成操作。

## 4.6 网络拓扑

本章节主要介绍在网络拓扑页面中，针对网络拓扑的一系列运维管理操作，如：查看网络拓扑、开/关设备标签显示、开/关资源折叠显示等。其中，在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络拓扑]，即可进入“网络拓扑”页面。

### 查看网络拓扑

在“网络拓扑”页面中，可查看当前云平台的网络拓扑结构图。单击图中资源或网络设备的图标，还可查看该资源或网络设备的ID和运行状态等信息。

### 开/关设备标签显示

在“网络拓扑”页面中，单击 **设备标签开关** ，可控制各资源或网络设备的名称是否显示。

### 开/关资源折叠显示

在“网络拓扑”页面中，单击 **资源折叠显示开关** ，可控制资源是否折叠显示。

# 5 最佳实践

## 5.1 通过Keepalived与虚拟IP结合配置云主机网络高可用

### 背景描述

通过为多个云主机的多个虚拟网卡配置同一虚拟IP（VIP），可以自定义其作为高可用服务的可漂移网络入口，保障业务应用的高可用性。典型的应用场景是将此虚拟IP与Keepalived相结合使用。

本文将以在两个CentOS 7云主机中配置Keepalived虚拟IP为例，详细介绍如何在该平台中配置Keepalived与虚拟IP结合使用，以保障业务连续性。

### 前提条件

- 已参考“计算”帮助中“云主机”的相关内容，完成两个CentOS 7云主机的创建。

### 操作步骤

1. 分别为各云主机的虚拟网卡配置同一虚拟IP。

- 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[虚拟网卡]，进入“虚拟网卡”页面。
- 勾选待操作虚拟网卡后， **更多** - **管理虚拟IP** ，弹出“管理虚拟IP”对话框。
- 单击 **添加IP** ，弹出“添加虚拟IP”对话框。配置参数后，单击 **保存** ，完成操作。

2. 在各云主机中安装Keepalived。

- 下载keepalived的rpm安装包。具体命令如下::

```
wget
http://www.rpmfind.net/linux/centos/7.6.1810/os/x86_64/Packages/keepalived-1.3.5-6.el7.x86_64.rpm
```



## 2. 配置EPEL源。

通过yum命令，配置EPEL源。具体命令如下::

```
yum -y
install http://dl.fedoraproject.org/pub/epel/7Server/x86_64/e/epel-
release-7-8.noarch.rpm
```

## 3. 安装keepalived的rpm安装包。具体命令如下::

```
yum localinstall keepalived-1.3.5-6.el7.x86_64.rpm -y
```

## 3. 配置各云主机的Keepalived。

1. 通过VIM编辑器，打开并编辑各云主机的Keepalived配置文件（即/etc/keepalived/keepalived.conf文件）。

Master云主机Instance A（**node 1**）的配置示例::

```
global_defs {
    router_id rt1
}
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    unicast_peer {
        $node2    #Fixed IP for Instance B.
    }
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1234
    }
    virtual_ipaddress {
        $vip dev eth0    #Use the VIP address you configured.
    }
}
```

Backup云主机Instance B (**node 2**) 的配置示例::

```
global_defs {
    router_id rt1
}
vrrp_instance VI_1 {
    state BACKUP
    interface eth0
    unicast_peer {
        $node1    #Fixed IP for Instance A.
    }
    virtual_router_id 51
    priority 80
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1234
    }
    virtual_ipaddress {
        $vip dev eth0    #Use the VIP address you configured.
    }
}
```

2. 启动Keepalived服务。具体命令如下::

```
service keepalived start
```

## 结果验证

1. 在Master云主机 (**node 1**) 中, 查询其IP信息, 确认虚拟IP已配置成功。查询IP信息的具体命令如下::

```
ip a
```

2. 停止Master云主机 (**node 1**) 的Keepalived服务。具体命令如下::

```
service keepalived stop
```

3. 查询Master云主机 (**node 1**) 和Backup云主机 (**node 2**) 的IP信息, 确认虚拟IP成功漂移到Backup云主机 (**node 2**) 上。

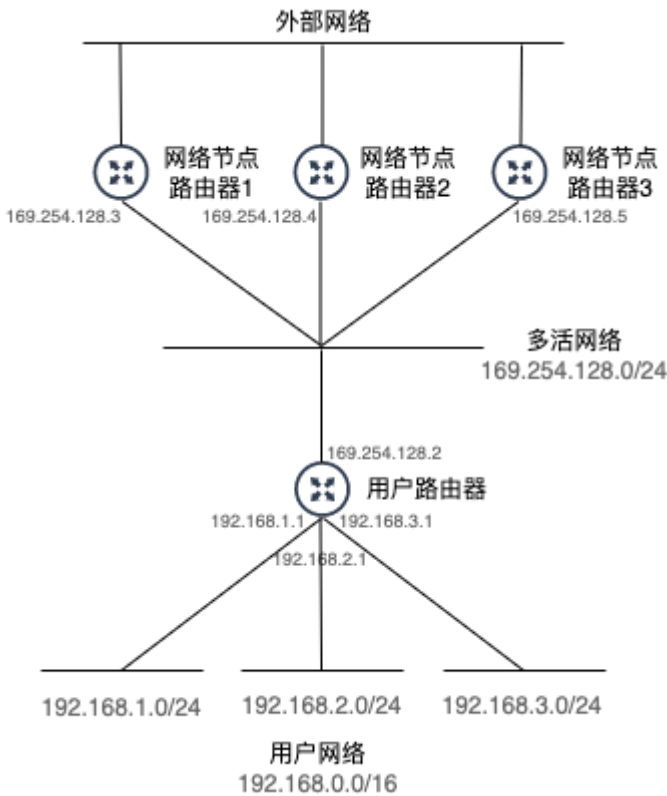
## 5.2 配置多活网络方案

### 背景描述

通过配置多活网络，可以横向提升网络性能，保障业务应用的高性能访问。本文将以配置三个网络节点路由器为例，介绍如何在该平台中配置多活网络方案，以提升业务的网络性能。

说明：

多活网络方案仅支持纯路由模式的路由器，即此方案中使用的用户路由器不支持公网IP和SNAT功能。



本实践方案中，具体网络规划信息如下：

类型	说明
----	----

类型	说明
用户网络	<ul style="list-style-type: none"> <li>* 名称: user_network</li> <li>* 网络类型: 内部网络</li> <li>* 网络模式: Geneve</li> <li>* 子网1:                             <ul style="list-style-type: none"> <li>- 名称: user_network_default_ipv4_subnet</li> <li>- 网段: 192.168.1.0/24</li> <li>- 网关地址: 设置网关 (192.168.1.1)</li> <li>- DHCP服务: 开启</li> </ul> </li> <li>* 子网2:                             <ul style="list-style-type: none"> <li>- 名称: user_network_default_ipv4_subnet2</li> <li>- 网段: 192.168.2.0/24</li> <li>- 网关地址: 设置网关 (192.168.2.1)</li> <li>- DHCP服务: 开启</li> </ul> </li> <li>* 子网3:                             <ul style="list-style-type: none"> <li>- 名称: user_network_default_ipv4_subnet3</li> <li>- 网段: 192.168.3.0/24</li> <li>- 网关地址: 设置网关 (192.168.3.1)</li> <li>- DHCP服务: 开启</li> </ul> </li> </ul>
多活网络	<ul style="list-style-type: none"> <li>* 名称: multi_net_169_254_128</li> <li>* 网络类型: 内部网络</li> <li>* 网络模式: Geneve</li> <li>* 子网:                             <ul style="list-style-type: none"> <li>- 名称: multi_net_169_254_128_default_ipv4_subnet</li> <li>- 网段: 169.254.128.0/24</li> <li>- 网关地址: 不设置</li> <li>- DHCP服务: 关闭</li> </ul> </li> </ul>

类型	说明
用户路由器	* 名称: user_router * 路由器连接: - 子网: multi_net_169_254_128_default_ipv4_subnet:169.254.128.0/24; 子网IP: 169.254.128.2 - 子网: user_network_default_ipv4_subnet:192.168.1.0/24; 子网IP: 192.168.1.1 - 子网: user_network_default_ipv4_subnet2:192.168.2.0/24; 子网IP: 192.168.2.1 - 子网: user_network_default_ipv4_subnet3:192.168.3.0/24; 子网IP: 192.168.3.1 * 静态路由: - 目的CIDR: 0.0.0.0/0; 下一跳: 169.254.128.3 - 目的CIDR: 0.0.0.0/0; 下一跳: 169.254.128.4 - 目的CIDR: 0.0.0.0/0; 下一跳: 169.254.128.5
网络节点路由器 1	* 名称: multi_ex_router_lb_1 * 可用区: 与网络节点的可用区一致 * 路由器网关: 172.16.10.3 (不开启SNAT) * 路由器连接: - 子网: multi_net_169_254_128_default_ipv4_subnet:169.254.128.0/24; 子网IP: 169.254.128.3 * 静态路由: - 目的CIDR: 192.168.0.0/16; 下一跳: 169.254.128.2
网络节点路由器 2	* 名称: multi_ex_router_lb_2 * 可用区: 与网络节点的可用区一致 * 路由器网关: 172.16.10.4 (不开启SNAT) * 路由器连接: - 子网: multi_net_169_254_128_default_ipv4_subnet:169.254.128.0/24; 子网IP: 169.254.128.4 * 静态路由: - 目的CIDR: 192.168.0.0/16; 下一跳: 169.254.128.2

类型	说明
网络节点路由器 3	<ul style="list-style-type: none"><li>* 名称: multi_ex_router_lb_3</li><li>* 可用区: 与网络节点的可用区一致</li><li>* 路由器网关: 172.16.10.5 (不开启SNAT)</li><li>* 路由器连接:<ul style="list-style-type: none"><li>- 子网: multi_net_169_254_128_default_ipv4_subnet:169.254.128.0/24; 子网IP: 169.254.128.5</li></ul></li><li>* 静态路由:<ul style="list-style-type: none"><li>- 目的CIDR: 192.168.0.0/16; 下一跳: 169.254.128.2</li></ul></li></ul>

## 操作步骤

### 1. 创建用户网络和多活网络。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次创建用户网络和多活网络。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，进入“网络”页面。
2. 单击 **创建网络** ，进入“创建网络”页面。
3. 配置参数后，单击 **创建网络** ，完成操作。其中，各参数的具体说明，请参考 [创建二层基础网络](#)。

### 2. 创建用户路由器和网络节点路由器。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次创建用户路由器和网络节点路由器。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[路由器]，进入“路由器”页面。
2. 单击 **创建路由器** ，弹出“创建路由器”对话框。
3. 配置参数后，单击 **创建** ，完成操作。其中，各参数的具体说明，请参考 [创建路由器](#)。

### 3. 设置网络节点路由器的网关。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置各网络节点路由器的网关。

1. 在“路由器”页面中，勾选待操作的网络节点路由器后，单击 **更多** - **设置网关** ，弹出“设置路由器网关”对话框。

2. 配置参数后，单击 **设置** ，完成操作。其中，各参数的具体说明，请参考 [设置路由器网关](#)。

4. 设置用户路由器和网络节点路由器的子网连接。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置用户路由器和网络节点路由器的子网连接。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。

2. 在[路由器连接]页签中，单击 **连接子网** ，弹出“连接子网”对话框。

3. 配置参数后，单击 **连接** ，完成操作。

5. 设置用户路由器和网络节点路由器的静态路由。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置用户路由器和网络节点路由器的静态路由。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。

2. 在[静态路由]页签中，单击 **添加静态路由** ，弹出“添加静态路由”对话框。

3. 配置参数后，单击 **连接** ，完成操作。

6. 设置外网网关的ECMP静态路由。

请在外部网关处，设置从外网网关回内网的静态路由。以Linux服务器网关为例，说明本实践方案中的静态路由配置命令::

```
ip route add 192.168.1.0/24 nexthop via 172.16.10.3 weight 1 nexthop via 172.16.10.4 weight 1 nexthop via 172.16.10.5 weight 1
ip route add 192.168.2.0/24 nexthop via 172.16.10.3 weight 1 nexthop via 172.16.10.4 weight 1 nexthop via 172.16.10.5 weight 1
ip route add 192.168.3.0/24 nexthop via 172.16.10.3 weight 1 nexthop via 172.16.10.4 weight 1 nexthop via 172.16.10.5 weight 1
```

## 结果验证



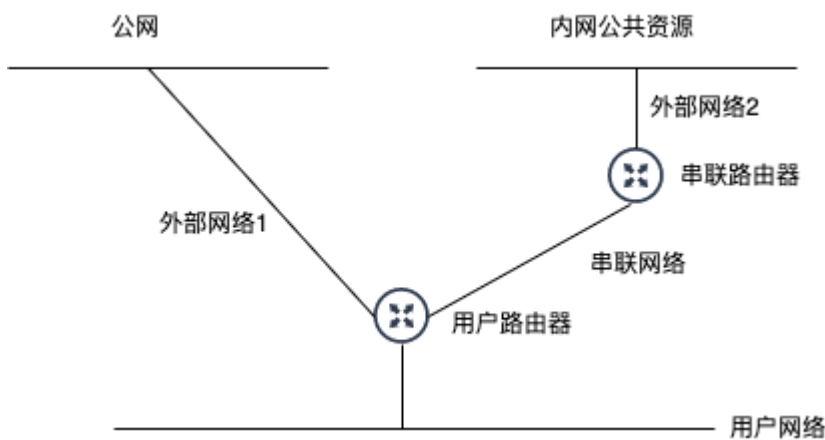
通过查看当前云平台的网络拓扑结构，确认各设备与资源之间的结构关系。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络拓扑]，进入“网络拓扑”页面。
2. 在“网络拓扑”页面中，查看当前云平台的网络拓扑结构图。

## 5.3 配置路由器连接多个外网

### 背景描述

通过配置路由器连接多个外部网络，可以实现云主机、容器或裸金属主机等计算资源访问安全隔离的多个网络环境，保障业务应用的安全可靠访问。本文将配置一个用户路由器连接两个外部网络（一个用于连接公网，一个用于连接内网公共资源）为例，介绍如何在该平台中配置路由器连接多个外部网络，以提升业务的高可靠性。



本实践方案中，具体网络规划信息如下：

类型	配置	说明
用户网络	<ul style="list-style-type: none"> <li>* 名称：user_network</li> <li>* 网络类型：内部网络</li> <li>* 网络模式：Geneve</li> <li>* 子网：                             <ul style="list-style-type: none"> <li>- 名称：user_network_default_ipv4_subnet</li> <li>- 网段：10.0.0.0/24</li> <li>- 网关地址：设置网关（10.0.0.1）</li> <li>- DHCP服务：开启</li> </ul> </li> </ul>	用于建立用户路由器与用户的云主机、容器或裸金属主机等计算资源之间的连接。

类型	配置	说明
外部网络1	<ul style="list-style-type: none"> <li>* 名称: ex_net_1</li> <li>* 网络类型: 外部网络</li> <li>* 网络模式: VLAN</li> <li>* 子网:                             <ul style="list-style-type: none"> <li>- 名称: ex_net_1_default_ipv4_subnet</li> <li>- 网段: 172.17.0.0/24</li> <li>- 网关地址: 设置网关 (172.17.0.1)</li> </ul> </li> </ul>	用于建立用户路由器与公网之间的连接。
外部网络2	<ul style="list-style-type: none"> <li>* 名称: ex_net_2</li> <li>* 网络类型: 外部网络</li> <li>* 网络模式: VLAN</li> <li>* 子网:                             <ul style="list-style-type: none"> <li>- 名称: ex_net_2_default_ipv4_subnet</li> <li>- 网段: 172.16.0.0/24</li> <li>- 网关地址: 设置网关 (172.16.0.1)</li> </ul> </li> </ul>	用于建立串联路由器与内网公共资源 (如对象存储、公共rpm源等) 之间的连接。
串联网络	<ul style="list-style-type: none"> <li>* 名称: series_network</li> <li>* 网络类型: 内部网络</li> <li>* 网络模式: Geneve</li> <li>* 子网:                             <ul style="list-style-type: none"> <li>- 名称: series_network_default_ipv4_subnet</li> <li>- 网段: 169.254.128.0/24</li> <li>- 网关地址: 不设置</li> <li>- DHCP服务: 关闭</li> </ul> </li> </ul>	用于建立用户路由器与串联路由器之间的连接。
用户路由器	<ul style="list-style-type: none"> <li>* 名称: user_router</li> <li>* 路由器网关: 172.17.0.10 (开启SNAT)</li> <li>* 路由器连接:                             <ul style="list-style-type: none"> <li>- 子网: user_network_default_ipv4_subnet:10.0.0.0/24; 子网IP: 10.0.0.1</li> <li>- 子网: series_network_default_ipv4_subnet:169.254.128.0/24; 子网IP: 169.254.128.2</li> </ul> </li> <li>* 静态路由:                             <ul style="list-style-type: none"> <li>- 目的CIDR: 172.16.0.0/24; 下一跳: 169.254.128.3</li> </ul> </li> </ul>	用于建立用户网络、公网与内网公共资源之间的连接。

类型	配置	说明
串联路由器	<ul style="list-style-type: none"><li>* 名称: series_router</li><li>* 可用区: 如需保证两个外部网络的访问物理隔离, 请指定单独的可用区, 并确保该可用区已配置独立的网络节点。</li><li>* 路由器网关: 172.16.0.10 (开启SNAT)</li><li>* 路由器连接:<ul style="list-style-type: none"><li>- 子网: series_network_default_ipv4_subnet: 169.254.128.0/24; 子网IP: 169.254.128.3</li></ul></li><li>* 静态路由:<ul style="list-style-type: none"><li>- 目的CIDR: 10.0.0.0/24; 下一跳: 169.254.128.2</li></ul></li></ul>	用于建立用户网络与内网公共资源(如对象存储、公共rpm源等)之间的连接。

## 操作步骤

### 1. 创建用户网络、外部网络和串联网络。

请依据本实践方案中的网络规划信息, 参考以下操作步骤, 依次创建用户网络、外部网络1、外部网络2和串联网络。

1. 在云平台的顶部导航栏中, 依次选择[产品与服务]-[网络]-[网络], 进入“网络”页面。
2. 单击 **创建网络**, 进入“创建网络”页面。
3. 配置参数后, 单击 **创建网络**, 完成操作。其中, 各参数的具体说明, 请参考 [创建二层基础网络](#)。

### 2. 创建用户路由器和串联路由器。

请依据本实践方案中的网络规划信息, 参考以下操作步骤, 依次创建用户路由器和串联路由器。

1. 在云平台的顶部导航栏中, 依次选择[产品与服务]-[网络]-[路由器], 进入“路由器”页面。
2. 单击 **创建路由器**, 弹出“创建路由器”对话框。
3. 配置参数后, 单击 **创建**, 完成操作。其中, 各参数的具体说明, 请参考 [创建路由器](#)。

### 3. 设置串联路由器的网关。

请依据本实践方案中的网络规划信息，参考以下操作步骤，设置串联路由器的网关。

1. 在“路由器”页面中，勾选待操作的网络节点路由器后，单击 **更多** - **设置网关**，弹出“设置路由器网关”对话框。
  2. 配置参数后，单击 **设置**，完成操作。其中，各参数的具体说明，请参考 [设置路由器网关](#)。
4. 设置用户路由器和串联路由器的子网连接。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置用户路由器和串联路由器的子网连接。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。
  2. 在[路由器连接]页签中，单击 **连接子网**，弹出“连接子网”对话框。
  3. 配置参数后，单击 **连接**，完成操作。
5. 设置用户路由器的静态路由。

请依据本实践方案中的网络规划信息，参考以下操作步骤，设置用户路由器的静态路由。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。
2. 在[静态路由]页签中，单击 **添加静态路由**，弹出“添加静态路由”对话框。
3. 配置参数后，单击 **连接**，完成操作。

## 结果验证

通过查看当前云平台的网络拓扑结构，确认各设备与资源之间的结构关系。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络拓扑]，进入“网络拓扑”页面。
2. 在“网络拓扑”页面中，查看当前云平台的网络拓扑结构图。

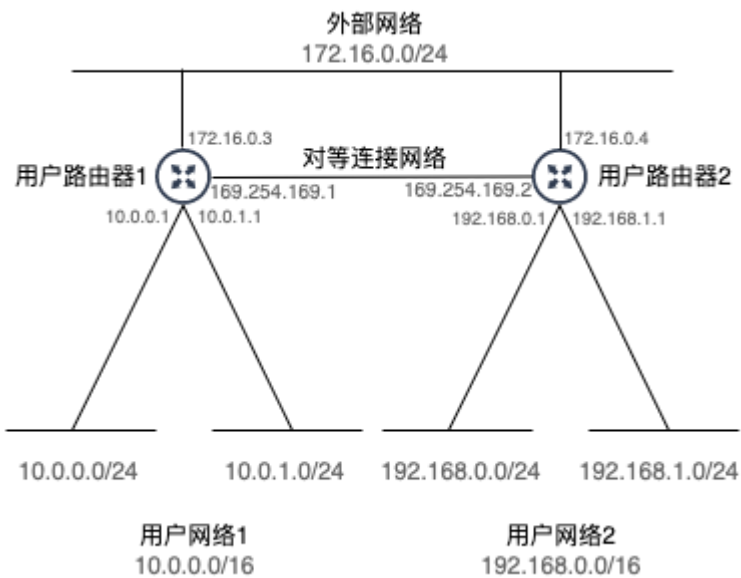
## 5.4 配置云内对等连接

### 背景描述

通过配置用户内部网络中多个路由器之间的对等连接，可以实现云内路由器下各子网之间的三层网络互通。本文将配置两个用户路由器的对等连接为例，介绍如何在该平台中配置内部网络的对等连接，实现云内三层网络互通。

**警告：**

在配置多个用户路由器之间的云内对等连接时，请确保各路由器的子网不发生冲突。



本实践方案中，具体网络规划信息如下（其中，用户网络与外部网络假定已完成相关配置，具体信息如上图所示，下表仅展示需另添加的信息）：

类型	配置	说明
----	----	----

类型	配置	说明
对等连接网络	<ul style="list-style-type: none"><li>* 名称: peer_network</li><li>* 网络类型: 内部网络</li><li>* 网络模式: Geneve</li><li>* 子网:<ul style="list-style-type: none"><li>- 名称: peer_network_default_ipv4_subnet</li><li>- 网段: 169.254.169.0/29</li><li>- 网关地址: 不设置</li><li>- DHCP服务: 关闭</li></ul></li></ul>	用于建立两个用户路由器之间的对等连接。
用户路由器1	<ul style="list-style-type: none"><li>* 名称: user_router_1</li><li>* 路由器连接:<ul style="list-style-type: none"><li>- 子网: peer_network_default_ipv4_subnet:169.254.169.0/29; 子网IP: 169.254.169.1</li></ul></li><li>* 静态路由:<ul style="list-style-type: none"><li>- 目的CIDR: 192.168.0.0/16; 下一跳: 169.254.169.2</li></ul></li></ul>	用于建立用户网络、外部网络与其他内部网络之间的连接。
用户路由器2	<ul style="list-style-type: none"><li>* 名称: user_router_2</li><li>* 路由器连接:<ul style="list-style-type: none"><li>- 子网: peer_network_default_ipv4_subnet:169.254.169.0/29; 子网IP: 169.254.169.2</li></ul></li><li>* 静态路由:<ul style="list-style-type: none"><li>- 目的CIDR: 10.0.0.0/16; 下一跳: 169.254.169.1</li></ul></li></ul>	用于建立用户网络、外部网络与其他内部网络之间的连接。

## 操作步骤

### 1. 创建对等连接网络。

请依据本实践方案中的网络规划信息，参考以下操作步骤，创建对等连接网络。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，进入“网络”页面。
2. 单击 **创建网络** ，进入“创建网络”页面。

3. 配置参数后，单击 **创建网络** ，完成操作。其中，各参数的具体说明，请参考 [创建二层基础网络](#)。

## 2. 创建用户路由器。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次创建用户路由器1和用户路由器2。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[路由器]，进入“路由器”页面。

2. 单击 **创建路由器** ，弹出“创建路由器”对话框。

3. 配置参数后，单击 **创建** ，完成操作。其中，各参数的具体说明，请参考 [创建路由器](#)。

## 3. 设置用户路由器的子网连接。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置用户路由器1和用户路由器2的子网连接。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。

2. 在[路由器连接]页签中，单击 **连接子网** ，弹出“连接子网”对话框。

3. 配置参数后，单击 **连接** ，完成操作。

## 4. 设置用户路由器的静态路由。

请依据本实践方案中的网络规划信息，参考以下操作步骤，依次设置用户路由器1和用户路由器2的静态路由。

1. 在“路由器”页面中，单击待操作路由器的名称，进入其详情页面。

2. 在[静态路由]页签中，单击 **添加静态路由** ，弹出“添加静态路由”对话框。

3. 配置参数后，单击 **连接** ，完成操作。

# 结果验证

通过查看当前云平台的网络拓扑结构，确认各设备与资源之间的结构关系。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络拓扑]，进入“网络拓扑”页面。

2. 在“网络拓扑”页面中，查看当前云平台的网络拓扑结构图。



# 6 常见问题

## 6.1 云主机IP地址未提供服务，如何排查解决

### 问题描述

在云平台中，可以查看到云主机的网络已为其分配虚拟网卡和IP地址，但该IP地址并不提供服务，而且其他设备也无法通过该IP地址访问到该云主机。

### 问题原因

该云主机连接的网络未开启DHCP功能。

### 解决方案

#### 1. 开启DHCP服务。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[网络]-[网络]，进入“网络”页面。
2. 单击上述云主机网络的名称后，进入其详情页面。在“子网”区域框中，勾选待操作子网后，单击 **更多** - **编辑** ，进入“编辑子网”页面。
3. 展开“高级配置”参数后，配置DHCP服务为“开启”。单击 **保存** ，完成操作。

#### 2. 重启云主机。

1. 在云平台的顶部导航栏中，依次选择[产品与服务]-[计算]-[云主机]，进入“云主机”页面。
2. 勾选上述云主机后，单击 **重启** ，弹出“重启云主机”提示框。
3. 单击 **重启** ，完成操作。

# 7 API参考

## 7.1 API简介

欢迎使用API文档，如果您熟悉网络服务协议和一种以上编程语言，推荐您调用API管理您的资源和开发自己的应用程序。本文档提供了API的描述、语法、参数说明及示例等内容。在调用API之前，请确保已经充分了解相关术语，详细信息请参见下表。

术语	说明
网络	网络与现实世界的交换机 / 路由器 / 服务器 / 连线组成的基础设施网络类似，创建网络后，用户可以在网络内创建子网，创建云主机时选择网络，组建服务器集群。我们提供的基础网络包含共享网络和外部网络，创建在共享网络上的云主机处于同一个网络内，通过安全组保障云主机访问安全。外部网络主要用于公网IP地址的分配。用户可以为项目创建内部网络，并在内部网络中创建子网。如同在物理网络上通过交换机将服务器连接到一起的局域网，服务器通过交换机连接到子网中。不同的内部网络之间是完全隔离的，因此不同的网络中可以配置相同的IP地址而不会产生冲突。同一个网络内可以创建多个子网，以适应业务的需求。
子网	子网是一个IPv4或IPv6的地址段。每个子网都需要定义IP地址的范围和掩码。子网和网络是多对1的关系，一个子网只能属于某个网络；一个网络可以有多个子网，这些子网可以是不同的IP段，但不能重叠，但如果子网在不同的网络中，网段是可以重叠的。
虚拟网卡	虚拟网卡可以看做是一个虚拟网口。虚拟网卡上定义了MAC地址和IP地址。虚拟网卡和子网是多对1的关系。一个虚拟网卡必须属于某个子网，一个子网可以有多个虚拟网卡。
QoS策略	QoS（Quality of Service，服务质量），可以用于设置虚拟网卡的带宽限制。
安全组	安全组提供网络间的访问控制功能，通过安全组中的安全组规则对当前项目中的网络流量进行过滤。安全组规则支持多种网络协议。

术语	说明
路由器	用户创建路由器，为不同的子网提供三层路由，从而让子网内的云主机与其他子网的云主机互联互通。也可以将用户创建的内部网络连接到外部网络，让内部网络的云主机访问Internet。路由器配置网关后，还可以为内网的云主机做端口转发，以节约公网IP地址资源。
公网IP	独立的IP地址资源，用户可以将申请的公网IP绑定到自己的云主机上，之后便可从外部网络通过公网IP来访问云主机提供的服务。
网络拓扑	展示用户当前所在项目的网络结构图。点击各个设备可以展示详细配置。
云主机	运行在云环境上的虚拟机，相当于数据中心的一台物理服务器。用户可以通过选择合适的CPU / 内存 / 操作系统磁盘空间，网络，安全组等配置创建云主机。
部门	部门是云平台中用户权限的一个划分层级，用户不能横跨多个部门。
项目	项目是定义资源所有权的基本单元，可理解为租户。所有资源（如云主机等）都要隶属于某个项目中。项目必须隶属于一个部门。项目名称在单个部门中的管理范围内是唯一的，但在整个云平台中可以不唯一。
用户	用户可以被云管理员、部门管理员创建。用户通过登录后，可以操作云平台提供的各项资源，如云主机 / 云硬盘等。

## 7.2 调用方式

### 请求结构

API支持基于URI发起HTTP/HTTPS GET请求。请求参数需要包含在URI中。本文列举了GET请求中的结构解释，并以云主机的服务接入地址为例进行了说明。

### 结构示例

以下为一条未编码的URI请求示例：`http://cloud.com/v1/{project_id}/servers` 在本示例中：

- `http` 指定了请求通信协议
- `cloud.com` 指定了服务接入地址
- `/v1/{project_id}/servers` 为资源路径，也即API访问路径

### 通信协议

支持HTTP或HTTPS协议请求通信。为了获得更高的安全性，推荐您使用HTTPS协议发送请求。涉及敏感数据时，如用户密码和SSH密钥对，推荐使用HTTPS协议。

### 服务网址

调用本文档所列举的API时均需使用OpenStack身份服务进行身份验证。他们还需要一个从“compute”类型的标识符提取出来的“service URI”。这将是根URI，将添加下面的每个调用来构建一个完整的路径。例如，如果“service URI”是 `http://mycompute.pvt/compute/v2.1`，那么“/servers”的完整API调用是

`http://mycompute.pvt/compute/v2.1/servers`。根据部署计算服务网址可能是http或https，自定义端口，自定义路径，并包含您的租户ID。要知道您的部署网址的唯一方法是通过使用服务目录。计算URI不应该被硬编码在应用程序中，即使他们只希望在单一地点工作。应始终从身份令牌中发现。因此，对于本文件的其余部分，我们将使用短针，其中“GET /servers”的真正含义“GET your\_compute\_service\_URI/servers”。

### 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。

方法	说明
----	----

方法	说明
GET	从服务端读取指定资源的所有信息，包括数据内容和元数据（Metadata）信息，其中元数据在响应头（Response Header）中返回，数据内容在响应体（Response Body）中。
PUT	向指定的资源上传数据内容和元数据信息。如果资源已经存在，那么新上传的数据将覆盖之前的内容。
POST	向指定的资源上传数据内容。与PUT操作相比，POST的主要区别在于POST一般用来向原有的资源添加信息，而不是替换原有的内容：POST所指的资源一般是处理请求的服务，或是能够处理多块数据。
DELETE	请求服务器删除指定资源，如删除对象等。
HEAD	仅从服务端读取指定资源的元数据信息。

## 字符编码

请求及返回结果都使用UTF-8字符集编码。

## 公共参数

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

### 公共请求参数

名称	类型	是否必选	描述
Host	String	否（使用AK/SK认证时该字段必选）	请求的服务器信息，从服务API的URI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。

名称	类型	是否必选	描述
Content-Type	String	是	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。
Content-Length	String	否	请求body长度，单位为Byte。
X-Project-Id	String	否	project id，项目编号。
X-Auth-Token	String	否（使用Token认证时该字段必选）	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。

## 公共返回参数

参数名称	参数类型	描述
RequestId	String	请求ID。无论调用接口成功与否，都会返回该参数。

## 签名机制

调用接口的认证方式为Token认证，通过Token认证通用请求。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": {
    "scope": {
      "project": {
        "domain": {
          "name": "Default"
        }
      }
    }
  }
}
    
```

```
    },
    "name": "admin"
  }
},
"identity": {
  "password": {
    "user": {
      "password": "devstacker",
      "id": "858634b407e845f14b02bcf369225dcd0"
    }
  },
  "methods": ["password"]
}
}
```

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.mycloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

## 返回结果

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。为了便于查看和美观，API 文档返回示例均有换行和缩进等处理，实际返回结果无换行和缩进处理。

## 正确返回结果

接口调用成功后会返回接口返回参数和请求 ID，我们称这样的返回为正常返回。HTTP 状态码为 2xx。以云主机的接口创建云主机（POST `/v1/{project_id}/servers`）为例，若调用成功，其可能的返回如下：

```
{
  "error": {
    "OS-DCF:diskConfig": "AUTO",
    "adminPass": "6NpUwoz2QDRN",
```

```
    "id": "f5dc173b-6804-445a-a6d8-c705dad5b5eb",
    "links": [
      {
        "href":
"http://openstack.example.com/v2/6f70656e737461636b20342065766572/servers/f5
dc173b-6804-445a-a6d8-c705dad5b5eb",
        "rel": "self"
      },
      {
        "href":
"http://openstack.example.com/6f70656e737461636b20342065766572/servers/f5dc1
73b-6804-445a-a6d8-c705dad5b5eb",
        "rel": "bookmark"
      }
    ],
    "security_groups": [
      {
        "name": "default"
      }
    ]
  }
}
```

## 错误返回结果

接口调用出错后，会返回错误码、错误信息和请求 ID，我们称这样的返回为异常返回。HTTP 状态码为 4xx 或者 5xx。

```
{
  "error": {
    "message": "The request you have made requires authentication.",
    "code": 401,
    "title": "Unauthorized"
  }
}
```

## 公共错误码



http状态码	Error Message	说明
300	multiple choices	被请求的资源存在多个可供选择的响应。
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。
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。

## 7.3 网络

### 网络

#### 列举网络

##### 功能介绍

列出租户有权访问的网络。除具有管理权限的用户提交列举子网的请求外，默认策略是只返回提交请求的用户所拥有的网络。

##### URI

```
GET /v2.0/networks
```

##### 请求消息

无

##### 响应消息

参数	参数类型	描述
id	string	网络ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
admin_state_up	boolean	该网络是否启动。
port_security_enabled	boolean	该网络是否开启网络安全防护功能。如果使用安全组功能，此配置项一定为打开。
mtu	int	该网络数据包最大传输单元大小。
shared	boolean	该网络是否允许全资源池项目都可见。

参数	参数类型	描述
router:external	boolean	该网络是否为外部网络。
availability_zone_hints	array	该网络的可用区候补选项。
availability_zones	array	该网络的可用区。
subnets	array	该网络的状态。
provider:network_type	string	该网络的类型。例如flat、vlan、geneve。
provider:physical_network	string	该网络的segment来自哪个物理网络。
provider:segmentation_id	string	该网络的segmentation ID。
ipv4_address_scope	string	该网络关联的ipv4地址scope。
status	string	该网络的状态。
description	string	该网络的自定义描述。
tags	array	该网络的自定义标签。
updated_at	string	该网络最新更新时间。
created_at	string	该网络创建时间。
qos_policy_id	string	该网络绑定的qos策略ID。
revision_number	int	该网络的操作码。
ipv6_address_scope	string	该网络关联的ipv6地址scope。

## 请求示例

```
GET http://{endpoint}/v2.0/networks
```

## 正常响应示例

```
{
  "network": {
    "id": "553f02e3-b0d0-4f9a-a818-42e0c82448ba",
    "name": "test",
    "tenant_id": "5608307e73d845918339684a78503e78",
    "project_id": "5608307e73d845918339684a78503e78",
    "admin_state_up": true,
    "port_security_enabled": true,
    "mtu": 1500,
    "shared": false,
    "router:external": false,
    "availability_zone_hints": ["default-az"],
    "availability_zones": ["default-az"],
    "subnets": [
      "0ce8208f-8c50-4a10-92e6-894edc025e2a",
      "a6b9609c-aac5-4cf0-91dc0c-45b247de6228"
    ],
    "provider:network_type": "vlan",
    "provider:physical_network": "physnet2",
    "provider:segmentation_id": 1015,
    "ipv4_address_scope": null,
    "status": "ACTIVE",
    "description": "",
    "tags": [ ],
    "updated_at": "2021-02-19T03:54:41Z",
    "created_at": "2021-02-19T03:04:35Z",
    "qos_policy_id": null,
    "revision_number": 7,
    "ipv6_address_scope": null
  }
}
```

## 正常响应代码

200

## 错误码

401

## 创建网络

### 功能介绍

创建一个网络。

### URI

POST /v2.0/networks

### 请求消息

参数	参数类型	是否必选	Description
name	string	是	该网络的名称。默认是一个空字符串。
project_id	string	否	该网络所属的项目ID。
availability_zone_hints	array	否	该网络所属的可用区候选。
availability_zones	array	否	该网络所属的可用区。
shared	boolean	否	该网络是否全局共享。
router:external	boolean	否	该网络是否为外部网络。默认是内部网络。
provider:network_type	string	否	该网络的类型。例如flat、vlan、geneve。
provider:physical_network	string	否	该网络的segment来自哪个物理网络。
provider:segmentation_id	int	否	该网络的segmentation ID。
subnets	array	否	该网络中的子网。

参数	参数类型	是否必选	Description
tags	array	否	该网络的标签。

## 响应消息

参数	参数类型	描述
id	string	网络ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
admin_state_up	boolean	该网络是否启动。
port_security_enabled	boolean	该网络是否开启网络安全防护功能。如果使用安全组功能，此配置项一定为打开。
mtu	int	该网络数据包最大传输单元大小。
shared	boolean	该网络是否允许全资源池项目都可见。
router:external	boolean	该网络是否为外部网络。
availability_zone_hints	array	该网络的可用区候补选项。
availability_zones	array	该网络的可用区。
subnets	array	该网络的状态。
provider:network_type	string	该网络的类型。例如flat、vlan、geneve。
provider:physical_network	string	该网络的segment来自哪个物理网络。

参数	参数类型	描述
provider:segmentation_id	string	该网络的segmentation ID。
ipv4_address_scope	string	该网络关联的ipv4地址scope。
status	string	该网络的状态。
description	string	该网络的自定义描述。
tags	array	该网络的自定义标签。
updated_at	string	该网络最新更新时间。
created_at	string	该网络创建时间。
qos_policy_id	string	该网络绑定的qos策略ID。
revision_number	int	该网络的操作码。
ipv6_address_scope	string	该网络关联的ipv6地址scope。

## 请求示例

```
POST http://{endpoint}/v2.0/networks
```

```
{
  "network": {
    "name": "test"
  }
}
```

## 正常响应示例

```
{
  "network": {
    "id": "553f02e3-b0d0-4f9a-a818-42e0c82448ba",
    "name": "test",
    "tenant_id": "5608307e73d845918339684a78503e78",
    "project_id": "5608307e73d845918339684a78503e78",
    "admin_state_up": true,
    "port_security_enabled": true,
    "mtu": 1500,
    "shared": false,
    "router:external": false,
    "availability_zone_hints": ["default-az"],
    "availability_zones": ["default-az"],
    "subnets": [],
    "provider:network_type": "vlan",
    "provider:physical_network": "physnet2",
    "provider:segmentation_id": 1015,
    "ipv4_address_scope": null,
    "status": "ACTIVE",
    "description": "",
    "tags": [],
    "updated_at": "2021-02-19T03:54:41Z",
    "created_at": "2021-02-19T03:04:35Z",
    "qos_policy_id": null,
    "revision_number": 7,
    "ipv6_address_scope": null
  }
}
```

## 正常响应代码

201

## 错误码

400, 401

## 显示网络详情

## 功能介绍



显示某个网络的详细信息。

## URI

```
GET /v2.0/networks/{network_id}
```

参数	参数类型	是否必选	Description
network_id	string	是	获取详情的network的ID。

## 请求消息

无

## 响应消息

参数	参数类型	描述
id	string	网络ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
admin_state_up	boolean	该网络是否启动。
port_security_enabled	boolean	该网络是否开启网络安全防护功能。如果使用安全组功能，此配置项一定为打开。
mtu	int	该网络数据包最大传输单元大小。
shared	boolean	该网络是否允许全资源池项目都可见。
router:external	boolean	该网络是否为外部网络。
availability_zone_hints	array	该网络的可用区候补选项。

参数	参数类型	描述
availability_zones	array	该网络的可用区。
subnets	array	该网络的状态。
provider:network_type	string	该网络的类型。例如flat、vlan、geneve。
provider:physical_network	string	该网络的segment来自哪个物理网络。
provider:segmentation_id	string	该网络的segmentation ID。
ipv4_address_scope	string	该网络关联的ipv4地址scope。
status	string	该网络的状态。
description	string	该网络的自定义描述。
tags	array	该网络的自定义标签。
updated_at	string	该网络最新更新时间。
created_at	string	该网络创建时间。
qos_policy_id	string	该网络绑定的qos策略ID。
revision_number	int	该网络的操作码。
ipv6_address_scope	string	该网络关联的ipv6地址scope。

## 请求示例

```
GET http://{endpoint}/v2.0/networks/553f02e3-b0d0-4f9a-a818-42e0c82448ba
```

## 正常响应示例

```
{
  "network": {
    "id": "553f02e3-b0d0-4f9a-a818-42e0c82448ba",
    "name": "test",
    "tenant_id": "5608307e73d845918339684a78503e78",
    "project_id": "5608307e73d845918339684a78503e78",
    "admin_state_up": true,
    "port_security_enabled": true,
    "mtu": 1500,
    "shared": false,
    "router:external": false,
    "availability_zone_hints": ["default-az"],
    "availability_zones": ["default-az"],
    "subnets": [],
    "provider:network_type": "vlan",
    "provider:physical_network": "physnet2",
    "provider:segmentation_id": 1015,
    "ipv4_address_scope": null,
    "status": "ACTIVE",
    "description": "",
    "tags": [],
    "updated_at": "2021-02-19T03:54:41Z",
    "created_at": "2021-02-19T03:04:35Z",
    "qos_policy_id": null,
    "revision_number": 7,
    "ipv6_address_scope": null
  }
}
```

## 正常响应代码

200

## 错误码

400, 401

## 更新网络

## 功能介绍

更新一个网络。

## URI

```
PUT /v2.0/networks/{network_id}
```

参数	参数类型	是否必选	Description
network_id	string	是	需要更新的network的ID。

## 请求消息

参数	参数类型	是否必选	Description
name	string	否	网络更新后的name。

## 响应消息

参数	参数类型	描述
id	string	网络ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
admin_state_up	boolean	该网络是否启动。
port_security_enabled	boolean	该网络是否开启网络安全防护功能。如果使用安全组功能，此配置项一定为打开。
mtu	int	该网络数据包最大传输单元大小。
shared	boolean	该网络是否允许全资源池项目都可见。
router:external	boolean	该网络是否为外部网络。

参数	参数类型	描述
availability_zone_hints	array	该网络的可用区候补选项。
availability_zones	array	该网络的可用区。
subnets	array	该网络的状态。
provider:network_type	string	该网络的类型。例如flat、vlan、geneve。
provider:physical_network	string	该网络的segment来自哪个物理网络。
provider:segmentation_id	string	该网络的segmentation ID。
ipv4_address_scope	string	该网络关联的ipv4地址scope。
status	string	该网络的状态。
description	string	该网络的自定义描述。
tags	array	该网络的自定义标签。
updated_at	string	该网络最新更新时间。
created_at	string	该网络创建时间。
qos_policy_id	string	该网络绑定的qos策略ID。
revision_number	int	该网络的操作码。
ipv6_address_scope	string	该网络关联的ipv6地址scope。

## 请求示例

```
PUT http://{endpoint}/v2.0/networks/553f02e3-b0d0-4f9a-a818-42e0c82448ba
```

```
{
  "network": {
    "name": "test-new"
  }
}
```

## 正常响应示例

```
{
  "network": {
    "id": "553f02e3-b0d0-4f9a-a818-42e0c82448ba",
    "name": "test-new",
    "tenant_id": "5608307e73d845918339684a78503e78",
    "project_id": "5608307e73d845918339684a78503e78",
    "admin_state_up": true,
    "port_security_enabled": true,
    "mtu": 1500,
    "shared": false,
    "router:external": false,
    "availability_zone_hints": ["default-az"],
    "availability_zones": ["default-az"],
    "subnets": [],
    "provider:network_type": "vlan",
    "provider:physical_network": "physnet2",
    "provider:segmentation_id": 1015,
    "ipv4_address_scope": null,
    "status": "ACTIVE",
    "description": "",
    "tags": [],
    "updated_at": "2021-02-19T03:54:41Z",
    "created_at": "2021-02-19T03:04:35Z",
    "qos_policy_id": null,
    "revision_number": 7,
    "ipv6_address_scope": null
  }
}
```

## 正常响应代码

---

200

## 错误码

400, 401, 403, 404, 412

## 删除网络

### 功能介绍

删除一个网络。

### URI

```
DELETE /v2.0/networks/{network_id}
```

参数	参数类型	是否必选	Description
network_id	string	是	需要删除网络的ID。

### 请求消息

无

### 响应消息

无

### 请求示例

```
DELETE http://{endpoint}/v2.0/networks/553f02e3-b0d0-4f9a-a818-42e0c82448ba
```

### 正常响应示例

无

### 正常响应代码

204

## 错误码

401, 404, 409, 412



## 7.4 子网

### 子网

#### 列举子网

##### 功能介绍

列出租户有权访问的子网。

##### URI

```
GET /v2.0/subnets
```

##### 请求消息

无

##### 响应消息

参数	参数类型	描述
subnets	array	子网对象的列表。
id	string	子网ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
enable_dhcp	boolean	子网是否开启或关闭dhcp功能。
network_id	string	子网所属的网络ID。
dns_nameservers	array	子网关联的dns服务器列表。

参数	参数类型	描述
allocation_pools	array	该子网的起始IP地址和结束IP地址分配池。
host_routes	array	子网的其他路由。带有目的地和下一跳参数的字典列表。
ip_version	integer	IP协议版本。取值为4 或 6。
gateway_ip	string	子网网关IP地址。如果该值为null，则表示该子网没有关联网关。
cidr	string	子网的CIDR。
created_at	string	创建资源的时间(UTC ISO8601格式)。
description	string	人类可读的资源描述。
ipv6_address_mode	string	IPv6地址分配用于指定IP地址分配的方式。取值为dhcpv6-stateful或null。
ipv6_ra_mode	string	IPv6路由器指定网络服务是否应该为一个子网发送ICMPv6报文。取值为dhcpv6-stateful或null。
revision_number	integer	资源的修订号。
service_types	array	子网关联的业务类型。
subnetpool_id	string	子网关联的子网池ID。
updated_at	string	更新资源的时间(UTC ISO8601格式)。

## 请求示例

```
GET http://{endpoint}/v2.0/subnets
```

## 正常响应示例

```
{  
  "subnets": [  
    ...  
  ]  
}
```

```
{
  "service_types": [
  ],
  "description": "",
  "enable_dhcp": true,
  "network_id": "553f02e3-b0d0-4f9a-a818-42e0c82448ba",
  "tenant_id": "5608307e73d845918339684a78503e78",
  "created_at": "2021-02-19T03:54:41Z",
  "dns_nameservers": [
  ],
  "updated_at": "2021-02-19T03:54:41Z",
  "gateway_ip": "192.168.1.1",
  "ipv6_ra_mode": null,
  "allocation_pools": [
    {
      "start": "192.168.1.2",
      "end": "192.168.1.254"
    }
  ],
  "host_routes": [
  ],
  "revision_number": 2,
  "ip_version": 4,
  "ipv6_address_mode": null,
  "cidr": "192.168.1.0/24",
  "project_id": "5608307e73d845918339684a78503e78",
  "id": "0ce8208f-8c50-4a10-92e6-894edc025e2a",
  "subnetpool_id": null,
  "name": "my_subnet_2"
},
{
  "service_types": [
  ],
  "description": "",
  "enable_dhcp": true,
  "network_id": "132fbb7d-a615-48c3-aa3f-7fc82970e2fb",
  "tenant_id": "fe0215103828465d925e0a4c2c81bfff1",
```

```
"created_at": "2021-02-19T10:34:42Z",
"dns_nameservers": [
],
"updated_at": "2021-02-19T10:34:43Z",
"gateway_ip": "10.100.0.225",
"ipv6_ra_mode": null,
"allocation_pools": [
  {
    "start": "10.100.0.226",
    "end": "10.100.0.238"
  }
],
"host_routes": [
],
"revision_number": 2,
"ip_version": 4,
"ipv6_address_mode": null,
"cidr": "10.100.0.224/28",
"project_id": "fe0215103828465d925e0a4c2c81bfff1",
"id": "0fd87cad-88ac-442c-af65-882a38b41c15",
"subnetpool_id": null,
"name": "my_subnet"
}
]
}
```

## 正常响应代码

200

## 错误码

401

## 创建子网

### 功能介绍

在网络中创建子网。

## URI

POST /v2.0/subnets

## 请求消息

参数	参数类型	是否必选	Description
subnet	object	是	一个子网对象。
tenant_id	string	否	资源所属项目的ID。只有管理员和具有advsvc角色的用户可以指定他们自己以外的项目ID。不能通过授权策略更改此值。
name	string	否	资源的人类可读名称。默认是一个空字符串。
enable_dhcp	boolean	否	子网是否开启或关闭dhcp功能。默认是true。
network_id	string	是	子网所属的网络ID。
allocation_pools	array	否	该子网的起始IP地址和结束IP地址池。如果不配置“allocation_pools”参数，将自动把CIDR中的所有IP地址分配到IP地址池，默认不包括子网网关预留地址。
host_routes	array	否	子网的其他路由。带有目的地和下一跳参数的字典列表。默认值为空列表。
ip_version	integer	是	IP协议版本。取值为4或6。
gateway_ip	string	否	子网网关IP地址。如果该值为null，则表示该子网没有关联网关。如果不指定gateway_ip，则会默认从CIDR中为子网分配网关地址。
cidr	string	是	子网的CIDR。
description	string	否	人类可读的资源描述。默认是一个空字符串。

参数	参数类型	是否必选	Description
ipv6_address_mode	string	否	IPv6地址分配用于指定IP地址分配的方式。取值为dhcpv6-stateful或null。
ipv6_ra_mode	string	否	IPv6路由器指定网络服务是否应该为一个子网发送ICMPv6报文。取值为dhcpv6-stateful或null。

## 响应消息

参数	参数类型	描述
subnet	object	一个子网对象。
id	string	子网ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
enable_dhcp	boolean	子网是否开启或关闭dhcp功能。
network_id	string	子网所属的网络ID。
dns_nameservers	array	子网关联的dns服务器列表。
allocation_pools	array	该子网的起始IP地址和结束IP地址池。
host_routes	array	子网的其他路由。带有目的地和下一跳参数的字典列表。
ip_version	integer	IP协议版本。取值为4 或 6。
gateway_ip	string	子网网关IP地址。如果该值为null，则表示该子网没有关联网关。
cidr	string	子网的CIDR。

参数	参数类型	描述
created_at	string	创建资源的时间(UTC ISO8601格式)。
description	string	人类可读的资源描述。
ipv6_address_mode	string	IPv6地址分配方式用于指定IP地址的分配方式。取值为dhcpv6-stateful或null。
ipv6_ra_mode	string	IPv6路由器指定网络服务是否应该为一个子网发送ICMPv6报文。取值为dhcpv6-stateful或null。
revision_number	integer	资源的修订号。
service_types	array	子网关联的业务类型。
subnetpool_id	string	子网关联的子网池ID。
updated_at	string	更新资源的时间(UTC ISO8601格式)。

## 请求示例

```
POST http://{endpoint}/v2.0/subnets
```

```
{
  "subnet": {
    "network_id": "605a9cd6-678f-44ed-9adc-5ad5eeeb9806",
    "ip_version": 4,
    "cidr": "192.168.4.0/24"
  }
}
```

## 正常响应示例

```
{
  "subnet": {
    "service_types": [
```

```
],
  "description": "",
  "enable_dhcp": true,
  "network_id": "605a9cd6-678f-44ed-9adc-5ad5eeeb9806",
  "tenant_id": "870c5414baee43a4ba7adfb253e46018",
  "created_at": "2021-02-24T02:24:55Z",
  "dns_nameservers": [

],
  "updated_at": "2021-02-24T02:24:55Z",
  "gateway_ip": "192.168.4.1",
  "ipv6_ra_mode": null,
  "allocation_pools": [
    {
      "start": "192.168.4.2",
      "end": "192.168.4.254"
    }
  ],
  "host_routes": [

],
  "revision_number": 2,
  "ip_version": 4,
  "ipv6_address_mode": null,
  "cidr": "192.168.4.0/24",
  "project_id": "43d9b6c8ee91457cb2e9bb86f701b3ef",
  "id": "8adac510-3352-43c2-bdca-dd017bf092ae",
  "subnetpool_id": null,
  "name": "my_subnet"
}
}
```

## 正常响应代码

201

## 错误码

400, 401, 403, 404, 409

## 显示子网详情



## 功能介绍

显示子网的详细信息。

## URI

```
GET /v2.0/subnets/{subnet_id}
```

参数	参数类型	是否必选	Description
subnet_id	string	是	

## 请求消息

无

## 响应消息

参数	参数类型	描述
subnet	object	一个子网对象。
id	string	子网ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
created_at	string	创建资源的时间(UTC ISO8601格式)。
name	string	资源的人类可读名称。
enable_dhcp	boolean	子网是否开启或关闭dhcp功能。
network_id	string	子网所属的网络ID。
dns_nameservers	array	子网关联的dns服务器列表。
allocation_pools	array	该子网的起始IP地址和结束IP地址池。

参数	参数类型	描述
host_routes	array	子网的其他路由。带有目的地和下一跳参数的字典列表。
ip_version	integer	IP协议版本。取值为4 或6。
gateway_ip	string	子网网关IP地址。如果该值为null, 则表示该子网没有关联网关。
cidr	string	子网的CIDR。
updated_at	string	更新资源的时间(UTC ISO8601格式)。
description	string	人类可读的资源描述。
ipv6_address_mode	string	IPv6地址分配用于指定IP地址分配的方式。取值为dhcpv6-stateful或null。
ipv6_ra_mode	string	IPv6路由器指定网络服务是否应该为一个子网发送ICMPv6报文。取值为dhcpv6-stateful或null。
revision_number	integer	资源的修订号。
service_types	array	子网关联的业务类型。
subnetpool_id	string	子网关联的子网池ID。

## 请求示例

```
GET http://{endpoint}/v2.0/subnets/008d3711-7978-439f-a340-707cb091aa85
```

## 正常响应示例

```
{
  "subnet": {
    "service_types": [
    ],
    "description": ""
  }
}
```

```
"enable_dhcp":true,
"network_id":"6c7ee1a2-25ad-4b83-98e6-f9412f679576",
"tenant_id":"870c5414baee43a4ba7adfb253e46018",
"created_at":"2021-02-24T02:34:49Z",
"dns_nameservers":[
],
"updated_at":"2021-02-24T02:34:49Z",
"gateway_ip":"192.168.1.1",
"ipv6_ra_mode":null,
"allocation_pools":[
  {
    "start":"192.168.1.2",
    "end":"192.168.1.254"
  }
],
"host_routes":[
],
"revision_number":2,
"ip_version":4,
"ipv6_address_mode":null,
"cidr":"192.168.1.0/24",
"project_id":"870c5414baee43a4ba7adfb253e46018",
"id":"008d3711-7978-439f-a340-707cb091aa85",
"subnetpool_id":null,
"name":"my_subnet"
}
}
```

## 正常响应代码

200

## 错误码

400, 401

## 更新子网

## 功能介绍

更新一个子网。某些属性，如IP版本（ip\_version）和CIDR（cidr）无法更新。尝试更新这些属性会导致“400 错误请求”错误。

## URI

```
PUT /v2.0/subnets/{subnet_id}
```

参数	参数类型	是否必选	Description
subnet_id	string	否	子网ID。

## 请求消息

参数	参数类型	是否必选	Description
name	string	否	资源的人类可读名称。
enable_dhcp	boolean	否	子网是否开启或关闭dhcp功能。默认是true。
dns_nameservers	array	否	子网关联的dns服务器列表。默认为空列表。
allocation_pools	array	否	该子网的起始IP地址和结束IP地址池。如果不配置“allocation_pools”参数，将自动把CIDR中的所有IP地址分配到IP地址池，默认不包括子网网关预留地址。
host_routes	array	否	子网的其他路由。带有目的地和下一跳参数的字典列表。默认值为空列表。
gateway_ip	string	否	子网网关IP地址。如果该值为null，则表示该子网没有关联网关。如果不指定gateway_ip，则默认从CIDR中为子网分配网关地址。

## 响应消息

参数	参数类型	描述
subnet	object	一个子网对象。
id	string	子网ID。
tenant_id	string	项目ID。
project_id	string	项目ID。
name	string	资源的人类可读名称。
enable_dhcp	boolean	子网是否开启或关闭dhcp功能。
network_id	string	子网所属的网络ID。
dns_nameservers	array	子网关联的dns服务器列表。
allocation_pools	array	该子网的起始IP地址和结束IP地址分配池。
host_routes	array	子网的其他路由。带有目的地和下一跳参数的字典列表。
ip_version	integer	IP协议版本。取值为4或6。
gateway_ip	string	子网网关IP地址。如果该值为null，则表示该子网没有关联网关。
cidr	string	子网的CIDR。
created_at	string	创建资源的时间(UTC ISO8601格式)。
description	string	人类可读的资源描述。
ipv6_address_mode	string	IPv6地址分配用于指定IP地址分配的方式。取值为dhcpv6-stateful或null。
ipv6_ra_mode	string	IPv6路由器指定网络服务是否应该为一个子网发送ICMPv6报文。取值为dhcpv6-stateful或null。
revision_number	integer	资源的修订号。

参数	参数类型	描述
service_types	array	子网关联的业务类型。
subnetpool_id	string	子网关联的子网池ID。
updated_at	string	更新资源的时间(UTC ISO8601格式)。

## 请求示例

```
PUT http://{endpoint}/v2.0/subnets/008d3711-7978-439f-a340-707cb091aa85
```

```
{
  "subnet": {
    "name": "my_subnet"
  }
}
```

## 正常响应示例

```
{
  "subnet": {
    "service_types": [],
    "description": "",
    "enable_dhcp": true,
    "network_id": "6c7ee1a2-25ad-4b83-98e6-f9412f679576",
    "tenant_id": "870c5414baee43a4ba7adfb253e46018",
    "created_at": "2021-02-24T02:34:49Z",
    "dns_nameservers": [],
    "updated_at": "2021-02-24T02:59:21Z",
    "gateway_ip": "192.168.1.1",
    "ipv6_ra_mode": null,
    "allocation_pools": [{
      "start": "192.168.1.2",
      "end": "192.168.1.254"
    }],
    "host_routes": [],
    "revision_number": 3,
  }
}
```

```
"ip_version": 4,  
"ipv6_address_mode": null,  
"cidr": "192.168.1.0/24",  
"project_id": "870c5414baee43a4ba7adfb253e46018",  
"id": "008d3711-7978-439f-a340-707cb091aa85",  
"subnetpool_id": null,  
"name": "my_subnet"  
}  
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 412

## 删除子网

### 功能介绍

删除一个子网。如果子网IP仍然处于被分配状态，则操作失败。

### URI

```
DELETE /v2.0/subnets/{subnet_id}
```

参数	参数类型	是否必选	Description
subnet_id	string	否	子网ID。

### 请求消息

无

### 响应消息

无

## 请求示例

```
DELETE http://{endpoint}/v2.0/subnets/008d3711-7978-439f-a340-707cb091aa85
```

## 正常响应示例

无

## 正常响应代码

204

## 错误码

401, 404, 412



## 7.5 虚拟网卡

### 虚拟网卡

#### 列举虚拟网卡

##### 功能介绍

列举租户有权限访问的网卡

##### URI

```
GET /v2.0/ports
```

##### 请求消息

参数	参数类型	是否必选	描述
fields	string	否	使用 <code>fields</code> 参数, API只返回请求的属性集,可以指定多次。例如, URL中指定 'fields=id&fields=name', 那么API只会返回'id'和'name'属性。
project_id	string	否	指定项目ID进行查询。

##### 响应消息

参数	参数类型	描述
ports	array	网卡列表。
admin_state_up	boolean	网卡的 admin 状态, 取值 true 或 false。
allowed_addresses_pairs	array	网卡允许访问的地址 pair 列表, 每个 pair 包含一个 ip 和 mac。

参数	参数类型	描述
binding:host_id	string	网卡所属的 host id。
binding:profile	object	网卡信息字典，传递和接收用于后端网络插件处理的网卡信息。
binding:vif_details	object	网卡附加信息字典，目前主要字段是 port_filter 和 ovs_hybrid_plug。
binding:vif_type	string	网卡绑定使用的 ml2 插件类型，比如 ovs, bridge, macvtap, 未绑定是 unbound, 失败是 binding_failed。
binding:vnic_type	string	网卡挂载的类型，比如 normal, direct, baremetal。
created_at	string	网卡创建时间 (in UTC ISO8601 format)。
description	string	用户描述。
device_id	string	使用该网卡的设备 ID，比如一个虚拟机 uuid。
device_owner	string	使用该网卡的设备类型，比如 network:dhcp。
extra_dhcp_opts	array	额外的 dhcp 参数列表。
fixed_ips	array	网卡的固定 ip 列表。
id	string	网卡的 ID。
mac_address	string	网卡的 mac 地址。
name	string	网卡的名称。
network_id	string	网卡所属的网络 ID。
port_security_enabled	boolean	网卡的安全状态。true 或者 false，如果是 true 将会配置安全组规则和 anti-spoofing 规则，否则不会处理。
project_id	string	网卡所属的 project id。
qos_policy_id	string	网卡关联的 qos 策略 id。

参数	参数类型	描述
revision_number	integer	网卡的版本号。
security_groups	array	网卡关联的安全组列表。
status	string	网卡的状态: ACTIVE, DOWN, BUILD and ERROR。
tags	array	网卡的标签列表。
qos_network_policy_id	string	网卡所属network绑定的qos policy id。
tenant_id	string	网卡所属的 tenant id。
updated_at	string	网卡最近更新的时间(in UTC ISO8601 format)。

## 请求示例

```
GET https://{endpoint}/v2.0/ports
```

## 正常响应示例

```
{
  "ports": [
    {
      "admin_state_up": true,
      "allowed_address_pairs": [],
      "binding:host_id": "node-10.domain.tld",
      "binding:profile": {},
      "binding:vif_details": {
        "ovs_hybrid_plug": false,
        "port_filter": true
      },
      "binding:vif_type": "ovs",
      "binding:vnic_type": "normal",
      "created_at": "2021-02-24T02:41:53Z",
      "description": "",
      "device_id": "f9583498-c576-4e2e-86ad-c6a662b000a8",
    }
  ]
}
```

```
"device_owner": "compute:kuryr",
"extra_dhcp_opts": [],
"fixed_ips": [
  {
    "ip_address": "192.168.111.12",
    "subnet_id": "f32a3ab8-6a5a-41c6-a202-cfe7bea9348a"
  }
],
"id": "04c4339b-d13f-4f14-9be5-8f7477cbea4e",
"mac_address": "fa:16:3e:f8:72:bb",
"name": "d717c61a2",
"network_id": "7b0343bd-cf33-45af-bb2b-25bea12eb2c9",
"port_security_enabled": true,
"project_id": "d717c61a263f4a0b9271d0a271653c2a",
"qos_policy_id": null,
"revision_number": 8,
"security_groups": [
  "a8c01103-2cf1-4f6a-a8f1-ed84ec2f794f"
],
"status": "ACTIVE",
"qos_network_policy_id": null,
"tenant_id": "d717c61a263f4a0b9271d0a271653c2a",
"updated_at": "2021-02-24T02:42:13Z"
}
]
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 创建虚拟网卡

### 功能介绍

创建一个虚拟网卡

## URI

POST /v2.0/ports

## 请求消息

参数	参数类型	是否必选	Description
port	object	是	网卡对象。
admin_state_up	boolean	否	网卡的admin状态, 取值true或false。
allowed_addresses_pairs	array	否	网卡允许访问的地址pair列表, 每个pair包含一个ip和mac。
binding:host_id	string	否	网卡所属的host id。
binding:profile	object	否	网卡信息字典, 传递和接收用于后端网络插件处理的网卡信息。
binding:vnic_type	string	否	网卡挂载的类型, 比如normal, direct, baremetal, 默认normal。
description	string	否	用户描述。
device_id	string	否	使用该网卡的设备ID, 比如一个虚拟机uuid。
device_owner	string	否	使用该网卡的设备类型, 比如network: DHCP。
extra_dhcp_options	array	否	额外的dhcp参数列表。
fixed_ips	array	否	网卡的固定ip列表。
mac_address	string	否	网卡的mac地址。
name	string	否	网卡的名称。
network_id	string	是	网卡所属的网络ID。

参数	参数类型	是否必选	Description
port_security_enabled	boolean	否	网卡的安全状态。true或者false，如果是true将会配置安全组规则和anti-spoofing规则，否则不会处理。
project_id	string	否	网卡所属的project id。
qos_policy_id	string	否	网卡关联的qos策略id。
security_groups	array	否	网卡关联的安全组列表。
tags	array	否	网卡的标签列表。
tenant_id	string	否	网卡所属的tenant id。

## 响应消息

参数	参数类型	描述
ports	array	网卡列表。
admin_state_up	boolean	网卡的 admin 状态，取值 true 或 false。
allowed_addresses_pairs	array	网卡允许访问的地址 pair 列表，每个 pair 包含一个 ip 和 mac。
binding:host_id	string	网卡所属的 host id。
binding:profile	object	网卡信息字典，传递和接收用于后端网络插件处理的网卡信息。
binding:vif_details	object	网卡附加信息字典，目前主要字段是 port_filter 和 ovs_hybrid_plug。
binding:vif_type	string	网卡绑定使用的 ml2 插件类型，比如 ovs, bridge, macvtap，未绑定是 unbound，失败是 binding_failed。
binding:vnic_type	string	网卡挂载的类型，比如 normal, direct, baremetal。

参数	参数类型	描述
created_at	string	网卡创建时间 (in UTC ISO8601 format)。
description	string	用户描述
device_id	string	使用该网卡的设备 ID, 比如一个虚拟机 uuid。
device_owner	string	使用该网卡的设备类型, 比如 network: DHC。
extra_dhcp_options	array	额外的dhcp参数列表。
fixed_ips	array	网卡的固定 ip 列表。
id	string	网卡的 ID。
mac_address	string	网卡的 mac 地址。
name	string	网卡的名称。
network_id	string	网卡所属的网络 ID。
port_security_enabled	boolean	网卡的安全状态。true 或者 false, 如果是 true 将会配置安全组规则和 anti-spoofing 规则, 否则不会处理。
project_id	string	网卡所属的 project id。
qos_policy_id	string	网卡关联的qos策略id。
revision_number	integer	网卡的版本号。
security_groups	array	网卡关联的安全组列表。
status	string	网卡的状态: ACTIVE, DOWN, BUILD and ERROR。
tags	array	网卡的标签列表。
qos_network_policy_id	string	网卡所属network绑定的qos policy id。
tenant_id	string	网卡所属的 tenant id。
updated_at	string	网卡最近更新的时间(in UTC ISO8601 format)。

## 请求示例

```
POST http://{endpoint}/v2.0/ports
```

```
{
  "port": {
    "admin_state_up": true,
    "name": "test",
    "network_id": "1191a2fd-42c6-44b9-b153-3378407468f2"
  }
}
```

## 正常响应示例

```
{
  "port": {
    "admin_state_up": true,
    "allowed_address_pairs": [],
    "binding:host_id": "",
    "binding:profile": {},
    "binding:vif_details": {},
    "binding:vif_type": "unbound",
    "binding:vnic_type": "normal",
    "created_at": "2021-02-24T07:01:14Z",
    "description": "",
    "device_id": "",
    "device_owner": "",
    "extra_dhcp_opts": [],
    "fixed_ips": [
      {
        "ip_address": "172.16.0.14",
        "subnet_id": "95347c6c-181d-4739-998d-0365487c9ac6"
      }
    ],
    "id": "602debb5-289d-4ecc-bc86-6a2f84a4b677",
    "mac_address": "fa:16:3e:91:6b:15",
    "name": "test",
    "network_id": "1191a2fd-42c6-44b9-b153-3378407468f2",
  }
}
```



```
"port_security_enabled": true,  
"project_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
"qos_policy_id": null,  
"revision_number": 5,  
"security_groups": [  
    "65db6fc3-7ba5-42e9-95f1-d45145f1e65f"  
],  
"status": "DOWN",  
"qos_network_policy_id": null,  
"tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
"updated_at": "2021-02-24T07:01:15Z"  
}  
}
```

## 正常响应代码

201

## 错误码

400, 401, 403, 404, 409, 413

## 获取虚拟网卡

### 功能介绍

获取一个虚拟网卡的详情

### URI

```
GET /v2.0/ports/{port_id}
```

参数	是否必选	描述
port_id	是	虚拟网卡的ID

### 请求消息

无

## 响应消息

参数	参数类型	描述
ports	array	网卡列表。
admin_state_up	boolean	网卡的 admin 状态, 取值 true 或 false。
allowed_addresses_pairs	array	网卡允许访问的地址 pair 列表, 每个 pair 包含一个 ip 和 mac。
binding:host_id	string	网卡所属的 host id。
binding:profile	object	网卡信息字典, 传递和接收用于后端网络插件处理的网卡信息。
binding:vif_details	object	网卡附加信息字典, 目前主要字段是 port_filter 和 ovs_hybrid_plug。
binding:vif_type	string	网卡绑定使用的 ml2 插件类型, 比如 ovs, bridge, macvtap, 未绑定是 unbound, 失败是 binding_failed。
binding:vnic_type	string	网卡挂载的类型, 比如 normal, direct, baremetal。
created_at	string	网卡创建时间 (in UTC ISO8601 format)。
description	string	用户描述。
device_id	string	使用该网卡的设备 ID, 比如一个虚拟机 uuid。
device_owner	string	使用该网卡的设备类型, 比如 network:distributed。
extra_dhcp_options	array	额外的 dhcp 参数列表。
fixed_ips	array	网卡的固定 ip 列表。
id	string	网卡的 ID。
mac_address	string	网卡的 mac 地址。
name	string	网卡的名称。

参数	参数类型	描述
network_id	string	网卡所属的网络 ID。
port_security_enabled	boolean	网卡的安全状态。true 或者 false，如果是 true 将会配置安全组规则和 anti-spoofing 规则，否则不会处理。
project_id	string	网卡所属的 project id。
qos_policy_id	string	网卡关联的qos策略id。
revision_number	integer	网卡的版本号。
security_groups	array	网卡关联的安全组列表。
status	string	网卡的状态：ACTIVE, DOWN, BUILD and ERROR。
tags	array	网卡的标签列表。
qos_network_policy_id	string	网卡所属network绑定的qos policy id。
tenant_id	string	网卡所属的 tenant id。
updated_at	string	网卡最近更新的时间(in UTC ISO8601 format)。

## 请求示例

```
GET https://{endpoint}/v2.0/ports/01a9c14a-d4bb-4b28-a5c8-fd0458b8c94e
```

## 正常响应示例

```
{
  "port": {
    "admin_state_up": false,
    "allowed_address_pairs": [],
    "binding:host_id": "",
    "binding:profile": {},
    "binding:vif_details": {},
  }
}
```

```
"binding:vif_type": "unbound",
"binding:vnic_type": "normal",
"created_at": "2021-02-22T12:29:22Z",
"description": "",
"device_id": "",
"device_owner": "",
"extra_dhcp_opts": [],
"fixed_ips": [
  {
    "ip_address": "172.16.0.9",
    "subnet_id": "c12ac01a-2422-473a-b297-baca99b2d690"
  }
],
"id": "01a9c14a-d4bb-4b28-a5c8-fd0458b8c94e",
"mac_address": "fa:16:3e:01:74:c7",
"name": "kw-2_kw-test_1edaecca",
"network_id": "328878f3-5805-42f1-82dd-40dcdd1f79f1",
"port_security_enabled": true,
"project_id": "8cd7d3e275584638964ff92ddd3b1596",
"qos_policy_id": null,
"revision_number": 5,
"security_groups": [
  "dbd9b12d-1d79-4594-ba18-36967bc4b9ae"
],
"status": "DOWN",
"qos_network_policy_id": null,
"tenant_id": "8cd7d3e275584638964ff92ddd3b1596",
"updated_at": "2021-02-22T12:29:22Z"
}
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 更新虚拟网卡

## 功能介绍

更新一个指定虚拟网卡

## URI

```
PUT /v2.0/ports/{port_id}
```

参数	是否必选	描述
port_id	是	虚拟网卡的ID

## 请求消息

参数	参数类型	是否必选	描述
port	object	是	网卡对象。
admin_state_up	boolean	否	网卡的admin状态，取值true或false。
allowed_addresses_pairs	array	否	网卡允许访问的地址pair列表，每个pair包含一个ip和mac。
binding:host_id	string	否	网卡所属的host id。
binding:profile	object	否	网卡信息字典，传递和接收用于后端网络插件处理的网卡信息。
binding:vnic_type	string	否	网卡挂载的类型，比如normal，direct，baremetal，默认normal。
description	string	否	用户描述。
device_id	string	否	使用该网卡的设备ID，比如一个虚拟机uuid。
device_owner	string	否	使用该网卡的设备类型，比如network:distributed。

参数	参数类型	是否必选	描述
extra_dhcp_opts	array	否	额外的dhcp参数列表。
fixed_ips	array	否	网卡的固定ip列表。
mac_address	string	否	网卡的mac地址。
name	string	否	网卡的名称。
port_security_enabled	boolean	否	网卡的安全状态。true或者false，如果是true将会配置安全组规则和anti-spoofing规则，否则不会处理。
qos_policy_id	string	否	网卡关联的qos策略id。
security_groups	array	否	网卡关联的安全组列表。

## 响应消息

参数	参数类型	描述
port	object	网卡对象。
admin_state_up	boolean	网卡的admin状态，取值true或false。
allowed_addresses_pairs	array	网卡允许访问的地址pair列表，每个pair包含一个ip和mac。
binding:host_id	string	网卡所属的host id。
binding:profile	object	网卡信息字典，传递和接收用于后端网络插件处理的网卡信息。
binding:vif_details	object	网卡附加信息字典，目前主要字段是port_filter和ovs_hybrid_plug。
binding:vif_type	string	网卡绑定使用的ml2插件类型，比如ovs，bridge，macvtap，未绑定是unbound，失败是binding_failed。

参数	参数类型	描述
binding_vnic_type	string	网卡挂载的类型, 比如normal, direct, baremetal。
created_at	string	网卡创建时间 (UTC ISO8601格式)。
description	string	用户描述。
device_id	string	使用该网卡的设备ID, 比如一个虚拟机uuid。
device_owner	string	使用该网卡的设备类型, 比如network: DHCP。
extra_dhcp_options	array	额外的dhcp参数列表。
fixed_ips	array	网卡的固定ip列表。
id	string	网卡的ID。
mac_address	string	网卡的mac地址。
name	string	网卡的名称。
network_id	string	网卡所属的网络ID。
port_security_enabled	boolean	网卡的安全状态。true或者false, 如果是true将会配置安全组规则和anti-spoofing规则, 否则不会处理。
project_id	string	网卡所属的project id。
qos_policy_id	string	网卡关联的qos策略id。
revision_number	integer	网卡的版本号。
security_groups	array	网卡关联的安全组列表。
status	string	网卡的状态: ACTIVE, DOWN, BUILD and ERROR。
tags	array	网卡的标签列表。
qos_network_policy_id	string	网卡所属network绑定的qos policy id

参数	参数类型	描述
tenant_id	string	网卡所属的tenant id。
updated_at	string	网卡最近更新的时间(UTC ISO8601 格式)。

## 请求示例

```
PUT http://{endpoint}/v2.0/ports/602debb5-289d-4ecc-bc86-6a2f84a4b677
```

```
{
  "port": {
    "fixed_ips": [
      {
        "ip_address": "172.16.0.115",
        "subnet_id": "95347c6c-181d-4739-998d-0365487c9ac6"
      }
    ]
  }
}
```

## 正常响应示例

```
{
  "port": {
    "admin_state_up": true,
    "allowed_address_pairs": [],
    "binding:host_id": "",
    "binding:profile": {},
    "binding:vif_details": {},
    "binding:vif_type": "unbound",
    "binding:vnic_type": "normal",
    "created_at": "2021-02-24T07:01:14Z",
    "description": "",
    "device_id": "",
    "device_owner": "",
    "extra_dhcp_opts": [],
    "fixed_ips": [
```



```
{
  "ip_address": "172.16.0.115",
  "subnet_id": "95347c6c-181d-4739-998d-0365487c9ac6"
},
{
  "id": "602debb5-289d-4ecc-bc86-6a2f84a4b677",
  "mac_address": "fa:16:3e:91:6b:15",
  "name": "test",
  "network_id": "1191a2fd-42c6-44b9-b153-3378407468f2",
  "port_security_enabled": true,
  "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
  "qos_policy_id": null,
  "revision_number": 8,
  "security_groups": [
    "65db6fc3-7ba5-42e9-95f1-d45145f1e65f"
  ],
  "status": "DOWN",
  "qos_network_policy_id": null,
  "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
  "updated_at": "2021-02-24T08:35:26Z"
}
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 删除虚拟网卡

### 功能介绍

删除一个指定虚拟网卡

### URI

```
DELETE /v2.0/ports/{port_id}
```

---

参数	是否必选	描述
port_id	是	虚拟网卡的ID

### 请求消息

无

### 请求示例

```
DELETE http://{endpoint}/v2.0/ports/01a9c14a-d4bb-4b28-a5c8-fd0458b8c94e
```

### 正常响应代码

204

### 错误码

400, 401, 403, 404, 409, 413

## 7.6 QoS策略

### QoS策略

#### 列举QoS policy

##### 功能介绍

列举QoS策略。

##### URI

```
GET /v2.0/qos/policies
```

##### 请求消息

无

##### 响应消息

参数	参数类型	描述
policies	array	policy对象列表
description	string	描述
tenant_id	string	租户id
project_id	string	租户id
revision_number	integer	版本号
shared	boolean	是否租户项目间共享
id	string	QoS策略id
is_default	boolean	是否是默认策略
rules	array	QoS策略规则列表

参数	参数类型	描述
name	string	名字
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
GET http://{endpoint}/v2.0/qos/policies
```

## 正常响应示例

```
{
  "policies": [
    {
      "name": "096c7e01-28ab-46e5-9427-1444f6f67a12",
      "rules": [
        {
          "max_kbps": 1024,
          "type": "bandwidth_limit",
          "id": "feaba4ba-c002-455c-810c-ffdaddac9c6a",
          "max_burst_kbps": 819,
          "qos_policy_id": "c15ce4a2-787e-4225-82f7-5ba538f0715e"
        }
      ],
      "tenant_id": "2caf92e24f284a76be9cb54577ecebed",
      "created_at": "2021-02-25T06:04:43Z",
      "updated_at": "2021-02-25T06:04:43Z",
      "revision_number": 2,
      "shared": false,
      "project_id": "2caf92e24f284a76be9cb54577ecebed",
      "id": "c15ce4a2-787e-4225-82f7-5ba538f0715e",
      "description": ""
    },
    {
      "name": "096c7e01-28ab-46e5-9427-1444f6f67a12",
      "rules": [
```

```
{
  "max_kbps":2048,
  "type":"bandwidth_limit",
  "id":"04cf4639-ffe8-42f3-bf79-42d22cccfcf2",
  "max_burst_kbps":1638,
  "qos_policy_id":"ffc9b0cd-f4db-4db3-ac82-9be3401e3a53"
},
{
  "tenant_id":"2caf92e24f284a76be9cb54577ecebed",
  "created_at":"2021-02-25T06:27:57Z",
  "updated_at":"2021-02-25T06:27:57Z",
  "revision_number":2,
  "shared":false,
  "project_id":"2caf92e24f284a76be9cb54577ecebed",
  "id":"ffc9b0cd-f4db-4db3-ac82-9be3401e3a53",
  "description":""
}
]
```

## 正常响应代码

200

## 错误码

401

## 创建QoS policy

### 功能介绍

创建QoS策略。

### URI

```
POST /v2.0/qos/policies
```

### 请求消息

参数	参数类型	是否必选	Description
policy	是	object	QoS策略对象
policy:name	是	string	QoS策略名字
policy:shared	否	bool	是否共享

## 响应消息

参数	参数类型	描述
policy	object	policy对象
description	string	描述
tenant_id	string	租户id
project_id	string	租户id
revision_number	integer	版本号
shared	boolean	是否租户项目间共享
id	string	QoS策略id
rules	array	QoS策略规则列表
name	string	名字
created_at	string	创建时间
updated_at	string	上一次更新时间
is_default	boolean	是否是默认策略

## 请求示例

```
POST http://{endpoint}/v2.0/qos/policies
```

```
{
  "policy":{
    "name":"test_qos"
  }
}
```

## 正常响应示例

```
{
  "policy":{
    "name":"test_qos",
    "rules":[

    ],
    "tenant_id":"cf8d21b7fc084813a0481fb9bf331357",
    "created_at":"2021-02-25T07:06:23Z",
    "updated_at":"2021-02-25T07:06:23Z",
    "revision_number":1,
    "shared":false,
    "project_id":"cf8d21b7fc084813a0481fb9bf331357",
    "id":"05e0ab13-0e6e-40d2-a04d-b5ead44d2653",
    "is_default": false,
    "description":""
  }
}
```

## 正常响应代码

201

## 错误码

401, 403, 404, 409

## 显示QoS policy详情

### 功能介绍

显示QoS策略详情。

## URI

```
GET /v2.0/qos/policies/{policy_id}
```

参数	参数类型	是否必选	Description
policy_id	string	是	policy_id

## 请求消息

无

## 响应消息

参数	参数类型	描述
policy	object	policy对象
description	string	描述
tenant_id	string	租户id
project_id	string	租户id
revision_number	integer	版本号
shared	boolean	是否租户项目间共享
id	string	QoS策略id
rules	array	QoS策略规则列表
name	string	名字
created_at	string	创建时间
updated_at	string	上一次更新时间
is_default	boolean	是否是默认策略



## 请求示例

```
GET http://{endpoint}/v2.0/qos/policies/ffc9b0cd-f4db-4db3-ac82-9be3401e3a53
```

## 正常响应示例

```
{
  "policy":{
    "name":"096c7e01-28ab-46e5-9427-1444f6f67a12",
    "rules":[
      {
        "max_kbps":2048,
        "type":"bandwidth_limit",
        "id":"04cf4639-ffe8-42f3-bf79-42d22ccccfcf2",
        "max_burst_kbps":1638,
        "qos_policy_id":"ffc9b0cd-f4db-4db3-ac82-9be3401e3a53"
      }
    ],
    "tenant_id":"2caf92e24f284a76be9cb54577ecebed",
    "created_at":"2021-02-25T06:27:57Z",
    "updated_at":"2021-02-25T06:27:57Z",
    "revision_number":2,
    "shared":false,
    "project_id":"2caf92e24f284a76be9cb54577ecebed",
    "id":"ffc9b0cd-f4db-4db3-ac82-9be3401e3a53",
    "is_default": false,
    "description":""
  }
}
```

## 正常响应代码

200

## 错误码

401, 404

## 给QoS policy设置带宽

## 功能介绍

给QoS策略设置带宽规则。

## URI

```
POST /v2.0/qos/policies/{policy_id}/bandwidth_limit_rules
```

参数	参数类型	是否必选	Description
policy_id	string	否	QoS policy id

## 请求消息

参数	参数类型	是否必选	Description
bandwidth_limit_rule	是	object	带宽限制对象
bandwidth_limit_rule:max_kbps	否	integer	每秒最大流量(K)
bandwidth_limit_rule:max_burst_kbps	否	integer	流量峰值(K)
bandwidth_limit_rule:direction	否	string	QoS规则方向 egress or ingress, 默认是 egress

## 响应消息

参数	参数类型	描述
bandwidth_limit_rule	object	带宽限制规则对象
max_kbps	integer	每秒最大数据量(K)
id	string	带宽规则id

参数	参数类型	描述
max_burst_kbps	integer	最大峰值流量(K)
direction	string	方向ingress(入口),egress(出口)

## 请求示例

```
POST http://{endpoint}/v2.0/qos/policies/05e0ab13-0e6e-40d2-a04d-b5ead44d2653/bandwidth_limit_rules
```

```
{
  "bandwidth_limit_rule": {
    "max_kbps": "2048",
    "max_burst_kbps": "2048"
  }
}
```

## 正常响应示例

```
{
  "bandwidth_limit_rule": {
    "id": "5f126d84-551a-4dcf-bb01-0e9c0df0c793",
    "max_kbps": 2048,
    "max_burst_kbps": 2048,
    "direction": "egress"
  }
}
```

## 正常响应代码

201

## 错误码

400, 401, 404, 409

## 更新QoS policy带宽

## 功能介绍

更新QoS带宽规则。

## URI

```
PUT /v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}
```

参数	参数类型	是否必选	Description
policy_id	string	是	QoS policy id
rule_id	string	是	QoS rule id

## 请求消息

参数	参数类型	是否必选	Description
bandwidth_limit_rule	是	object	带宽限制对象
bandwidth_limit_rule:max_kbps	否	integer	每秒最大流量(K)
bandwidth_limit_rule:max_burst_kbps	否	integer	流量峰值(K)
bandwidth_limit_rule:direction	否	string	QoS规则方向 egress or ingress, 默认是 egress

## 响应消息

参数	参数类型	描述
bandwidth_limit_rule	object	带宽限制规则对象
max_kbps	integer	每秒最大数据量(K)

参数	参数类型	描述
id	string	带宽规则id
max_burst_kbps	integer	最大峰值流量(K)
direction	string	方向ingress(入口),egress(出口)

## 请求示例

```
PUT http://{endpoint}/v2.0/qos/policies/05e0ab13-0e6e-40d2-a04d-b5ead44d2653/bandwidth_limit_rules/d1a8dc46-34df-453c-908c-a4eb36e0ba4f
```

```
{"bandwidth_limit_rule":{  
  "max_kbps":"2048",  
  "max_burst_kbps":"3096"  
}}
```

## 正常响应示例

```
{  
  "bandwidth_limit_rule":{  
    "max_kbps":2048,  
    "id":"d1a8dc46-34df-453c-908c-a4eb36e0ba4f",  
    "max_burst_kbps":3096,  
    "direction":"ingress"  
  }  
}
```

## 正常响应代码

200

## 错误码

400, 401, 404

## 删除QoS policy带宽规则

### 功能介绍

删除QoS带宽规则。

### URI

```
DELETE /v2.0/qos/policies/{policy_id}/bandwidth_limit_rules/{rule_id}
```

参数	参数类型	是否必选	Description
policy_id	string	是	QoS policy id
rule_id	string	是	QoS rule id

### 请求消息

无

### 响应消息

无

### 请求示例

```
DELETE https://{endpoint}/v2.0/qos/policies/05e0ab13-0e6e-40d2-a04d-b5ead44d2653/bandwidth_limit_rules/d1a8dc46-34df-453c-908c-a4eb36e0ba4f
```

### 正常响应示例

无

### 正常响应代码

204

### 错误码

400, 401, 404

## 7.7 路由器

### 路由器

#### 列举路由器

#### 功能介绍

列举路由器

#### URI

```
GET /v2.0/routers
```

#### 请求消息

参数	参数类型	是否必选	描述
project_id	string	否	指定项目ID进行查询。

#### 响应消息

参数	参数类型	描述
routers	array	router对象列表
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态, up/down(true/false)

参数	参数类型	描述
status	string	路由器状态
external_gateway_info	object	路由网关信息，需要有network_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号
routes	array	静态路由配置，内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
GET https://{endpoint}/v2.0/routers
```

## 正常响应示例

```
{
  "routers": [
    {
      "status": "ACTIVE",
      "external_gateway_info": null,
      "availability_zone_hints": [
        "default-az"
      ],
      "availability_zones": [
        "default-az"
      ],
      "description": ""
    }
  ]
}
```



```
    "admin_state_up":true,  
    "tenant_id":"870c5414baee43a4ba7adfb253e46018",  
    "created_at":"2021-02-24T07:53:09Z",  
    "updated_at":"2021-02-24T07:53:20Z",  
    "project_id":"870c5414baee43a4ba7adfb253e46018",  
    "revision_number":8,  
    "routes":[  
  
    ],  
    "id":"95405096-26e5-4f68-a952-5533c5f08292",  
    "name":"test_router"  
  }  
]  
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 创建路由器

### 功能介绍

创建一个路由器

### URI

POST /v2.0/routers

### 请求消息

参数	参数类型	是否必选	描述
router	object	是	路由器对象。

参数	参数类型	是否必选	描述
admin_state_up	boolean	否	路由器的admin状态，取值true或false。
availability_zone_hints	array	否	路由可用域范围
description	string	否	用户描述。
name	string	否	路由器的名称。
project_id	string	否	路由器所属的project id。
tenant_id	string	否	路由器所属的tenant id。

## 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态, up/down(true/false)
status	string	路由器状态
external_gateway_info	object	路由网关信息，需要有network_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号

参数	参数类型	描述
routes	array	静态路由配置，内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
POST http://{endpoint}/v2.0/routers
```

```
{
  "router": {
    "admin_state_up": true,
    "name": "test_router",
    "availability_zone_hints": [
      "default-az"
    ]
  }
}
```

## 正常响应示例

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": null,
    "availability_zone_hints": [
      "default-az"
    ]
  }
}
```

```
    ],
    "availability_zones": [
      "default-az"
    ],
    "description": "",
    "admin_state_up": true,
    "tenant_id": "870c5414baee43a4ba7adfb253e46018",
    "created_at": "2021-02-24T07:53:09Z",
    "updated_at": "2021-02-24T07:53:20Z",
    "project_id": "870c5414baee43a4ba7adfb253e46018",
    "revision_number": 8,
    "routes": [
      ],
    "id": "95405096-26e5-4f68-a952-5533c5f08292",
    "name": "test_router"
  }
}
```

## 正常响应代码

201

## 错误码

400, 401, 403, 404, 409, 413

## 获取路由器

### 功能介绍

获取一个路由器的详情

### URI

```
GET /v2.0/routers/{router_id}
```

参数	是否必选	描述
router_id	是	路由器的ID

## 请求消息

无

## 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态, up/down(true/false)
status	string	路由器状态
external_gateway_info	object	路由网关信息, 需要有network_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号
routes	array	静态路由配置, 内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
GET http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292
```

## 正常响应示例

```
{
  "router": {
    "status": "ACTIVE",
    "external_gateway_info": null,
    "availability_zone_hints": [
      "default-az"
    ],
    "availability_zones": [
      "default-az"
    ],
    "description": "",
    "admin_state_up": true,
    "tenant_id": "870c5414baee43a4ba7adfb253e46018",
    "created_at": "2021-02-24T07:53:09Z",
    "updated_at": "2021-02-24T07:53:20Z",
    "project_id": "870c5414baee43a4ba7adfb253e46018",
    "revision_number": 8,
    "routes": [
    ],
    "id": "95405096-26e5-4f68-a952-5533c5f08292",
    "name": "test_router"
  }
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 更新路由器

### 功能介绍

更新一个指定路由器

### URI

```
PUT /v2.0/routers/{router_id}
```

参数	是否必选	描述
router_id	是	路由器的ID

### 请求消息

参数	参数类型	是否必选	描述
router	object	是	路由器对象。
admin_state_up	boolean	否	路由器的admin状态，取值true或false。
description	string	否	用户描述。
name	string	否	路由器的名称。
routes	array	否	静态路由配置，内包含destination(目的地址)和nexthop(下一跳地址)

### 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id

参数	参数类型	描述
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态, up/down(true/false)
status	string	路由器状态
external_gateway_info	object	路由网关信息, 需要有netwrok_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号
routes	array	静态路由配置, 内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

添加静态路由

```
PUT http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292
```

```
{
  "router": {
    "routes": {
```



```
    "destination": "192.168.112.0/24",  
    "nexthop": "192.168.111.7"  
  }  
}  
}
```

## 正常响应示例

```
{  
  "router":  
    {  
      "status": "ACTIVE",  
      "external_gateway_info": null,  
      "availability_zone_hints": [  
        "default-az"  
      ],  
      "availability_zones": [  
        "default-az"  
      ],  
      "description": "",  
      "admin_state_up": true,  
      "tenant_id": "870c5414baee43a4ba7adfb253e46018",  
      "created_at": "2021-02-24T07:53:09Z",  
      "updated_at": "2021-02-24T07:53:20Z",  
      "project_id": "870c5414baee43a4ba7adfb253e46018",  
      "revision_number": 8,  
      "routes": [  
        {  
          "destination": "192.168.112.0/24",  
          "nexthop": "192.168.111.7"  
        }  
      ],  
      "id": "95405096-26e5-4f68-a952-5533c5f08292",  
      "name": "test_router"  
    }  
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 删除路由器

### 功能介绍

删除一个指定路由器

### URI

```
DELETE /v2.0/routers/{router_id}
```

参数	是否必选	描述
router_id	是	路由器的ID

### 请求消息

无

### 请求示例

```
DELETE http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292
```

### 正常响应代码

204

### 错误码

400, 401, 403, 404, 409, 413

## 路由器设置网关

### 功能介绍

给一个路由器设置外部网关

## URI

```
PUT /v2.0/routers/{router_id}
```

参数	是否必选	描述
router_id	是	路由器的ID

## 请求消息

参数	参数类型	是否必选	描述
router	object	是	路由器对象。
external_gateway_info	object	是	外部网络信息，必须指定network_id，可选项external_fixed_ips网关ip，rate_limit带宽，enable_snat是否开启snat默认开启

## 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态，up/down(true/false)

参数	参数类型	描述
status	string	路由器状态
external_gateway_info	object	路由网关信息，需要有network_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号
routes	array	静态路由配置，内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
PUT http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292
```

```
{
  "router":{
    "external_gateway_info":{
      "network_id":"2dd8b394-fa34-44e8-9ea7-0e470ee5bcd9",
      "external_fixed_ips":"172.16.10.130",
      "rate_limit":1024000
    }
  }
}
```

## 正常响应示例

```
{
  "router":
  {
    "status": "ACTIVE",
    "external_gateway_info": {
      "enable_snat": true,
      "external_fixed_ips": [
        {
          "ip_address": "172.16.10.130",
          "subnet_id": "08fe297b-db20-4b46-9341-a6812c9f748f"
        }
      ],
      "network_id": "2dd8b394-fa34-44e8-9ea7-0e470ee5bcd9",
      "rate_limit": 1024000
    },
    "availability_zone_hints": [
      "default-az"
    ],
    "availability_zones": [
      "default-az"
    ],
    "description": "",
    "admin_state_up": true,
    "tenant_id": "870c5414baee43a4ba7adfb253e46018",
    "created_at": "2021-02-24T07:53:09Z",
    "updated_at": "2021-02-24T07:53:20Z",
    "project_id": "870c5414baee43a4ba7adfb253e46018",
    "revision_number": 8,
    "routes": [
      {
        "destination": "192.168.112.0/24",
        "nexthop": "192.168.111.7"
      }
    ],
    "id": "95405096-26e5-4f68-a952-5533c5f08292",
    "name": "test_router"
  }
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 路由器清除网关

### 功能介绍

清除一个路由器的外部网关

### URI

PUT /v2.0/routers/{router\_id}

参数	是否必选	描述
router_id	是	路由器的ID

### 请求消息

参数	参数类型	是否必选	描述
router	object	是	路由器对象。
external_gateway_info	object	是	外部网络信息，此API中传空字典。

### 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id

参数	参数类型	描述
tenant_id	string	租户id
project_id	string	租户id
name	string	名字
description	string	描述
admin_state_up	boolean	admin状态, up/down(true/false)
status	string	路由器状态
external_gateway_info	object	路由网关信息, 需要有netwrok_id, enable_snat和external_fixed_ip否则字段为空
revision_number	integer	路由器的版本号
routes	array	静态路由配置, 内包含destination(目的地址)和nexthop(下一跳地址)
availability_zone_hints	array	路由可用域范围
availability_zones	array	路由所在的可用域
created_at	string	创建时间
updated_at	string	上一次更新时间

## 请求示例

```
PUT http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292
```

```
{
  "router":{
    "external_gateway_info":{
    }
  }
}
```

```
}  
}
```

## 正常响应示例

```
{  
  "router":  
    {  
      "status": "ACTIVE",  
      "external_gateway_info": null,  
      "availability_zone_hints": [  
        "default-az"  
      ],  
      "availability_zones": [  
        "default-az"  
      ],  
      "description": "",  
      "admin_state_up": true,  
      "tenant_id": "870c5414baee43a4ba7adfb253e46018",  
      "created_at": "2021-02-24T07:53:09Z",  
      "updated_at": "2021-02-24T07:53:20Z",  
      "project_id": "870c5414baee43a4ba7adfb253e46018",  
      "revision_number": 8,  
      "routes": [  
        {  
          "destination": "192.168.112.0/24",  
          "nexthop": "192.168.111.7"  
        }  
      ],  
      "id": "95405096-26e5-4f68-a952-5533c5f08292",  
      "name": "test_router"  
    }  
}
```

## 正常响应代码

200

## 错误码



400, 401, 403, 404, 409, 413

## 路由器连接子网

### 功能介绍

给一个路由器连接子网

### URI

```
PUT /v2.0/routers/{router_id}/add_router_interface
```

参数	是否必选	描述
router_id	是	路由器的ID

### 请求消息

参数	参数类型	是否必选	描述
subnet_id	string	是(subnet_id或者port_id两个二选一)	连接的子网的uuid
port_id	string	是	子网中port的uuid

### 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
network_id	string	连接的子网所属网络的uuid

参数	参数类型	描述
subnet_id	string	连接的子网的uuid
subnet_ids	array	连接的子网uuid列表
port_id	string	子网中port的uuid

### 请求示例

```
PUT http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292/add_router_interface
```

```
{  
  "subnet_id": "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47"  
}
```

### 正常响应示例

```
{  
  "id": "95405096-26e5-4f68-a952-5533c5f08292",  
  "network_id": "a1bd29e8-9349-44a5-9fdf-39462c08d761",  
  "port_id": "2b32e183-17fc-4d2f-bea9-355803bd9bc3",  
  "subnet_id": "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47",  
  "subnet_ids": [  
    "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47"  
  ],  
  "tenant_id": "870c5414baee43a4ba7adfb253e46018"  
}
```

### 正常响应代码

200

### 错误码

400, 401, 403, 404, 409, 413

## 路由器断开子网

### 功能介绍

断开一个路由器的某个子网

### URI

```
PUT /v2.0/routers/{router_id}/remove_router_interface
```

参数	是否必选	描述
router_id	是	路由器的ID

### 请求消息

参数	参数类型	是否必选	描述
subnet_id	string	是(subnet_id或者port_id两个二选一)	连接的子网的uuid
port_id	string	是	子网中port的uuid

### 响应消息

参数	参数类型	描述
router	object	router对象
id	string	router id
tenant_id	string	租户id
project_id	string	租户id
network_id	string	连接的子网所属网络的uuid
subnet_id	string	连接的子网的uuid

参数	参数类型	描述
subnet_ids	array	连接的子网uuid列表
port_id	string	子网中port的uuid

## 请求示例

```
PUT http://{endpoint}/v2.0/routers/95405096-26e5-4f68-a952-5533c5f08292/remove_router_interface
```

```
{  
  "subnet_id": "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47"  
}
```

## 正常响应示例

```
{  
  "id": "95405096-26e5-4f68-a952-5533c5f08292",  
  "network_id": "a1bd29e8-9349-44a5-9fdf-39462c08d761",  
  "port_id": "2b32e183-17fc-4d2f-bea9-355803bd9bc3",  
  "subnet_id": "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47",  
  "subnet_ids": [  
    "8bb1d1e8-a479-4f4f-8ca1-3bc80eabfe47"  
  ],  
  "tenant_id": "870c5414baee43a4ba7adfb253e46018"  
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 7.8 安全组

### 安全组

#### 列举安全组

##### 功能介绍

列举网络安全组

##### URI

```
GET /v2.0/security-groups
```

##### 请求消息

参数	参数类型	是否必选	描述
project_id	string	否	指定项目ID进行查询。

##### 响应消息

参数	参数类型	描述
security-groups	array	列表组对象。
created_at	string	安全组的创建时间 (UTC ISO8601 格式)。
description	string	用户描述
id	string	安全组的ID。
name	string	安全组的名称。
project_id	string	安全组所属的project id。
revision_number	integer	安全组的版本号。

参数	参数类型	描述
security_group_rules	array	安全组包含的规则列表。
stateful	boolean	安全组是否为有状态。
tenant_id	string	安全组所属的tenant id。
updated_at	string	安全组最近更新的时间(UTC ISO8601格式)。

### 请求示例

```
GET http://{endpoint}/v2.0/security-groups
```

### 正常响应示例

```
{
  "security_groups": [
    {
      "created_at": "2021-02-24T09:06:32Z",
      "description": "",
      "id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
      "name": "test",
      "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
      "revision_number": 1,
      "security_group_rules": [
        {
          "created_at": "2021-02-24T09:06:32Z",
          "description": null,
          "direction": "egress",
          "ethertype": "IPv4",
          "id": "231354cc-c516-47fd-9cc5-c20aab9b0e6c",
          "port_range_max": null,
          "port_range_min": null,
          "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
          "protocol": null,
          "remote_group_id": null,
          "remote_ip_prefix": null,
          "revision_number": 1,

```

```
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-24T09:06:32Z"
  },
  {
    "created_at": "2021-02-24T09:06:32Z",
    "description": null,
    "direction": "egress",
    "ethertype": "IPv6",
    "id": "2455c213-22ec-4d8d-b668-483cf8f831b3",
    "port_range_max": null,
    "port_range_min": null,
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "protocol": null,
    "remote_group_id": null,
    "remote_ip_prefix": null,
    "revision_number": 1,
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-24T09:06:32Z"
  }
],
"stateful": true,
"tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
"updated_at": "2021-02-24T09:06:32Z"
}
]
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 创建安全组

## 功能介绍

创建一个安全组

## URI

POST /v2.0/security-groups

## 请求消息

参数	参数类型	是否必选	Description
	security-group	object	是
description	string	否	用户描述。
name	string	否	安全组的名称。
stateful	boolean	否	安全组是否为有状态。
tenant_id	string	否	所属租户ID。

## 响应消息

参数	参数类型	描述
security-group	object	安全组对象。
created_at	string	安全组的创建时间 (UTC ISO8601 格式)。
description	string	用户描述。
id	string	安全组的ID。
name	string	安全组的名称。
project_id	string	安全组所属的project id。
revision_number	integer	安全组的版本号。
security_group_rules	array	安全组包含的规则列表。
stateful	boolean	安全组是否为有状态。



参数	参数类型	描述
tenant_id	string	安全组所属的tenant id。
updated_at	string	安全组最近更新的时间(UTC ISO8601格式)。

## 请求示例

```
POST http://{endpoint}/v2.0/security-groups
```

```
{
  "security_group": {
    "name": "test",
    "stateful": true,
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d"
  }
}
```

## 正常响应示例

```
{
  "security_group": {
    "created_at": "2021-02-24T09:06:32Z",
    "description": "",
    "id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "name": "test",
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "revision_number": 1,
    "security_group_rules": [
      {
        "created_at": "2021-02-24T09:06:32Z",
        "description": null,
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "231354cc-c516-47fd-9cc5-c20aab9b0e6c",
        "port_range_max": null,
        "port_range_min": null,
        "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",

```

```
        "protocol": null,  
        "remote_group_id": null,  
        "remote_ip_prefix": null,  
        "revision_number": 1,  
        "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",  
        "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
        "updated_at": "2021-02-24T09:06:32Z"  
    },  
    {  
        "created_at": "2021-02-24T09:06:32Z",  
        "description": null,  
        "direction": "egress",  
        "ethertype": "IPv6",  
        "id": "2455c213-22ec-4d8d-b668-483cf8f831b3",  
        "port_range_max": null,  
        "port_range_min": null,  
        "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
        "protocol": null,  
        "remote_group_id": null,  
        "remote_ip_prefix": null,  
        "revision_number": 1,  
        "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",  
        "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
        "updated_at": "2021-02-24T09:06:32Z"  
    }  
],  
    "stateful": true,  
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",  
    "updated_at": "2021-02-24T09:06:32Z"  
}  
}
```

## 正常响应代码

201

## 错误码

400, 401, 403, 404, 409, 413

## 获取安全组

## 功能介绍

获取一个安全组的详情

## URI

```
GET /v2.0/security-groups/{security-group-id}
```

参数	是否必选	描述
security-group-id	是	安全组 的ID

## 请求消息

无

## 响应消息

参数	参数类型	描述
security-group	object	安全组对象。
created_at	string	安全组的创建时间 (UTC ISO8601 格式)。
description	string	用户描述。
id	string	安全组的ID。
name	string	安全组的名称。
project_id	string	安全组所属的project id。
revision_number	integer	安全组的版本号。
security_group_rules	array	安全组包含的规则列表。
stateful	boolean	安全组是否为有状态。
tenant_id	string	安全组所属的tenant id。
updated_at	string	安全组最近更新的时间(UTC ISO8601 格式)。

## 请求示例

```
GET http://{endpoint}/v2.0/ports/01a9c14a-d4bb-4b28-a5c8-fd0458b8c94e
```

## 正常响应示例

```
{
  "security_group": {
    "created_at": "2021-02-24T09:06:32Z",
    "description": "",
    "id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "name": "test",
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "revision_number": 1,
    "security_group_rules": [
      {
        "created_at": "2021-02-24T09:06:32Z",
        "description": null,
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "231354cc-c516-47fd-9cc5-c20aab9b0e6c",
        "port_range_max": null,
        "port_range_min": null,
        "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "revision_number": 1,
        "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
        "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
        "updated_at": "2021-02-24T09:06:32Z"
      },
      {
        "created_at": "2021-02-24T09:06:32Z",
        "description": null,
        "direction": "egress",
        "ethertype": "IPv6",
        "id": "2455c213-22ec-4d8d-b668-483cf8f831b3",
        "port_range_max": null,
        "port_range_min": null,

```

```
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "protocol": null,
    "remote_group_id": null,
    "remote_ip_prefix": null,
    "revision_number": 1,
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-24T09:06:32Z"
  }
],
"stateful": true,
"tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
"updated_at": "2021-02-24T09:06:32Z"
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 更新安全组

### 功能介绍

更新一个指定安全组

### URI

```
PUT /v2.0/security-groups/{security-group-id}
```

参数	是否必选	描述
security-group-id	是	安全组的ID

### 请求消息

参数	参数类型	是否必选	描述
security-group	object	是	安全组对象。
description	string	否	用户描述。
stateful	boolean	否	安全组是否为有状态。
name	string	否	安全组的名称。

## 响应消息

参数	参数类型	描述
security-group	object	安全组对象。
created_at	string	安全组的创建时间 (UTC ISO8601 格式)。
description	string	用户描述。
id	string	安全组的ID。
name	string	安全组的名称。
project_id	string	安全组所属的project id。
revision_number	integer	安全组的版本号。
security_group_rules	array	安全组包含的规则列表。
stateful	boolean	安全组是否为有状态。
tenant_id	string	安全组所属的tenant id。
updated_at	string	安全组最近更新的时间(UTC ISO8601格式)。

## 请求示例

```
PUT http://{endpoint}/v2.0/security-groups/de80111e-4ddb-4b58-a295-0b32e9a4f963
```

```
{
  "security_group": {
    "name": "test2"
  }
}
```

## 正常响应示例

```
{
  "security_group": {
    "created_at": "2021-02-24T09:06:32Z",
    "description": "",
    "id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "name": "test2",
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "revision_number": 1,
    "security_group_rules": [
      {
        "created_at": "2021-02-24T09:06:32Z",
        "description": null,
        "direction": "egress",
        "ethertype": "IPv4",
        "id": "231354cc-c516-47fd-9cc5-c20aab9b0e6c",
        "port_range_max": null,
        "port_range_min": null,
        "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
        "protocol": null,
        "remote_group_id": null,
        "remote_ip_prefix": null,
        "revision_number": 1,
        "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
        "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
        "updated_at": "2021-02-24T09:06:32Z"
      },
      {
        "created_at": "2021-02-24T09:06:32Z",
        "description": null,
        "direction": "egress",
        "ethertype": "IPv6",
```

```
    "id": "2455c213-22ec-4d8d-b668-483cf8f831b3",
    "port_range_max": null,
    "port_range_min": null,
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "protocol": null,
    "remote_group_id": null,
    "remote_ip_prefix": null,
    "revision_number": 1,
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-24T09:06:32Z"
  }
],
"stateful": true,
"tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
"updated_at": "2021-02-24T10:06:32Z"
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 删除安全组

### 功能介绍

删除一个指定安全组

### URI

```
DELETE /v2.0/security-groups/{security-group-id}
```

参数	是否必选	描述
security-group-id	是	安全组的ID



## 请求消息

无

## 请求示例

```
DELETE http://{endpoint}/v2.0/security-groups/de80111e-4ddb-4b58-a295-0b32e9a4f963
```

## 正常响应代码

204

## 错误码

400, 401, 403, 404, 409, 413

## 列举安全组规则

### 功能介绍

列举所有安全组规则

### URI

```
GET /v2.0/security-group-rules
```

## 请求消息

无

## 响应消息

参数	参数类型	描述
security-group-rules	array	安全组规则列表。
created_at	string	安全组规则的创建时间 (UTC ISO8601格式)。
description	string	用户描述。

参数	参数类型	描述
direction	string	安全组规则作用的方向, ingress或egress
ethertype	string	匹配的以太网协议, 必需是IPv4或IPv6。
id	string	安全组规则的ID。
project_id	string	安全组规则所。属的project id。
protocol	string	匹配的IP协议类型, 比如ICMP type。
port_range_max	integer	匹配的协议端口号最大值,如果协议是ICMP, 则表示ICMP code。
port_range_min	integer	匹配的协议端口号最小值,如果协议是ICMP, 则表示ICMP code。
remote_ip_prefix	string	匹配的远端ip前缀。
remote_group_id	string	匹配的远端安全组的ID, 可以指定remote_group_id或者remote_ip_prefix。
revision_number	integer	安全组规则的版本号。
security_group_id	string	关联的安全组ID。
normalized_cidr	string	只读的字段, 规范化cidr的值
tenant_id	string	安全组规则所属的tenant id。
updated_at	string	安全组规则最近更新的时间(UTC ISO8601格式)。

## 请求示例

```
GET http://{endpoint}/v2.0/security-group-rules
```

## 正常响应示例

```
{
  "security_group_rules": [
    {
      "created_at": "2021-02-22T03:23:55Z",
      "description": null,
      "direction": "egress",
      "ethertype": "IPv4",
      "id": "00c1068e-0110-4cf2-8856-0e20268562fd",
      "port_range_max": null,
      "port_range_min": null,
      "project_id": "6e3cf7a9d62a4faf8162cacfc94126a9",
      "protocol": null,
      "remote_group_id": null,
      "remote_ip_prefix": null,
      "revision_number": 1,
      "normalized_cidr": null,
      "security_group_id": "61884e28-ca98-4e76-b0f3-6d5b3536bd7b",
      "tenant_id": "6e3cf7a9d62a4faf8162cacfc94126a9",
      "updated_at": "2021-02-22T03:23:55Z"
    },
    {
      "created_at": "2021-02-25T03:07:27Z",
      "description": "",
      "direction": "ingress",
      "ethertype": "IPv4",
      "id": "0184bfe7-f838-476f-ad19-6891215583f5",
      "port_range_max": 65535,
      "port_range_min": 1,
      "project_id": "e9f2f863c96841e49ba4e9946ebe9087",
      "protocol": "tcp",
      "remote_group_id": null,
      "remote_ip_prefix": "0.0.0.0/0",
      "revision_number": 1,
      "normalized_cidr": null,
      "security_group_id": "a12e08af-9e8d-461c-952d-3c6fbc1bebc",
      "tenant_id": "e9f2f863c96841e49ba4e9946ebe9087",
      "updated_at": "2021-02-25T03:07:27Z"
    }
  ]
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 创建安全组规则

### 功能介绍

创建一个安全组规则

### URI

```
POST /v2.0/security-group-rules
```

### 请求消息

参数	参数类型	是否必选	Description
security-group-rule	object	是	安全组规则对象。
description	string	否	用户描述。
direction	string	是	安全组规则作用的方向, ingress或egress。
ethertype	string	否	匹配的以太网协议类型, 必需是IPv4或IPv6。
port_range_max	integer	否	匹配的协议端口号最大值,如果协议是ICMP, 则表示ICMP code。
port_range_min	integer	否	匹配的协议端口号最小值,如果协议是ICMP, 则表示ICMP code。
protocol	string	否	匹配的IP协议类型, 比如ICMP type。

参数	参数类型	是否必选	Description
remote_ip_prefix	string	否	匹配的远端ip前缀。
remote_group_id	string	否	匹配的远端安全组ID, 可以指定remote_group_id或者remote_ip_prefix。
security_group_id	string	是	关联的安全组ID。
tenant_id	string	否	所属租户ID。

## 响应消息

参数	参数类型	描述
security-group-rule	object	安全组规则。
created_at	string	安全组规则的创建时间 (UTC ISO8601格式)。
description	string	用户描述。
direction	string	安全组规则作用的方向, ingress或egress。
ethertype	string	匹配的以太网协议, 必需是IPv4或IPv6。
id	string	安全组规则的ID。
project_id	string	安全组规则所属的project id。
protocol	string	匹配的IP协议类型, 比如ICMP type。
port_range_max	integer	匹配的协议端口号最大值,如果协议是ICMP, 则表示ICMP code。
port_range_min	integer	匹配的协议端口号最小值,如果协议是ICMP, 则表示ICMP code。

参数	参数类型	描述
remote_ip_prefix	string	匹配的远端ip前缀。
remote_group_id	string	匹配的远端安全组的ID, 可以指定remote_group_id或者remote_ip_prefix。
revision_number	integer	安全组规则的版本号。
security_group_id	string	关联的安全组ID。
normalized_cidr	string	只读的字段, 规范化cidr的值
tenant_id	string	安全组规则所属的tenant id。
updated_at	string	安全组规则最近更新的时间(UTC ISO8601格式)。

## 请求示例

```
POST http://{endpoint}/v2.0/security-group-rules
```

```
{
  "security_group_rule": {
    "direction": "ingress",
    "ethertype": "IPv4",
    "port_range_max": "88",
    "port_range_min": "22",
    "protocol": "tcp",
    "remote_ip_prefix": "0.0.0.0/24",
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963"
  }
}
```

## 正常响应示例

```
{
  "security_group_rule": {
    "created_at": "2021-02-25T06:19:33Z",
    "description": "",
    "direction": "ingress",
    "ethertype": "IPv4",
    "id": "5dc48c92-e097-4aa4-ae9b-88550dd0170d",
    "port_range_max": 88,
    "port_range_min": 22,
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "protocol": "tcp",
    "remote_group_id": null,
    "remote_ip_prefix": "0.0.0.0/24",
    "revision_number": 1,
    "normalized_cidr": null,
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-25T06:19:33Z"
  }
}
```

## 正常响应代码

201

## 错误码

400, 401, 403, 404, 409, 413

## 获取安全组规则

### 功能介绍

获取一个安全组规则的详情

### URI

```
GET /v2.0/security-group-rules/{security-group-rule-id}
```

参数	是否必选	描述
security-group-rule-id	是	安全组规则的ID

## 请求消息

无

## 响应消息

参数	参数类型	描述
security-group-rule	object	安全组规则对象。
created_at	string	安全组规则的创建时间 (UTC ISO8601 form)。
description	string	用户描述。
direction	string	安全组规则作用的方向, ingress或egress
ethertype	string	匹配的以太网协议, 必需是IPv4或IPv6
id	string	安全组规则的ID。
project_id	string	安全组规则所属的project id。
protocol	string	匹配的IP协议类型, 比如ICMP type。
port_range_max	integer	匹配的协议端口号最大值,如果协议是ICMP, 则表示ICMP code。
port_range_min	integer	匹配的协议端口号最小值,如果协议是ICMP, 则表示ICMP code。
remote_ip_prefix	string	匹配的远端ip前缀。
remote_group_id	string	匹配的远端安全组的ID, 可以指定remote_group_id或者remote_ip_prefix。



参数	参数类型	描述
revision_number	integer	安全组规则的版本号。
security_group_id	string	关联的安全组ID。
normalized_cidr	string	只读的字段，规范化cidr的值
tenant_id	string	安全组规则所属的tenant id。
updated_at	string	安全组规则最近更新的时间(UTC ISO8601格式)。

## 请求示例

```
GET http://{endpoint}/v2.0/security-group-rules/ 5dc48c92-e097-4aa4-ae9b-88550dd0170d
```

## 正常响应示例

```
{
  "security_group_rule": {
    "created_at": "2021-02-25T06:19:33Z",
    "description": "",
    "direction": "ingress",
    "ethertype": "IPv4",
    "id": "5dc48c92-e097-4aa4-ae9b-88550dd0170d",
    "port_range_max": 88,
    "port_range_min": 22,
    "project_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "protocol": "tcp",
    "remote_group_id": null,
    "remote_ip_prefix": "0.0.0.0/24",
    "revision_number": 1,
    "normalized_cidr": null,
    "security_group_id": "de80111e-4ddb-4b58-a295-0b32e9a4f963",
    "tenant_id": "8dac1bf1c3a24be18809f82e1f9a845d",
    "updated_at": "2021-02-25T06:19:33Z"
  }
}
```

```
}  
}
```

## 正常响应代码

200

## 错误码

400, 401, 403, 404, 409, 413

## 删除安全组规则

### 功能介绍

删除一个指定安全组规则

### URI

```
DELETE /v2.0/security-group-rules/{security-group-rule-id}
```

参数	是否必选	描述
security-group-rule-id	是	安全组的ID

### 请求消息

无

### 请求示例

```
DELETE http://{endpoint}/v2.0/security-group-rules/ 5dc48c92-e097-4aa4-ae9b-88550dd0170d
```

## 正常响应代码

204

## 错误码

400, 401, 403, 404, 409, 413

## 7.9 公网IP

### 公网IP

#### 列举公网IP

##### 功能介绍

列举符合条件的所有公网IP

##### URI

```
GET /v2.0/floatingips
```

##### 请求消息

参数	参数类型	是否必选	Description
router_id	string	否	指定路由器的ID进行查询
project_id	string	否	指定项目ID进行查询
floating_network_id	string	否	公网IP地址申请自外部网络的ID
port_id	string	否	绑定公网IP的port的ID
status	string	否	公网IP的运行状态
tags	string	否	公网IP的标签

##### 响应消息

参数	参数类型	描述
floatingips	array	公网IP列表
id	string	公网IP的ID

参数	参数类型	描述
tenant_id	string	公网IP所属的用户ID
floating_ip_address	string	公网IP的地址
floating_network_id	string	公网IP地址申请自外部网络的ID
router_id	string	绑定公网IP所属router的ID
port_id	string	绑定公网IP的port的ID
fixed_ip_addresses	string	绑定公网IP的port的IP地址
status	string	公网IP的运行状态
description	string	公网IP的描述
qos_policy_id	string	公网IP绑定的qos策略
port_details	string	绑定公网IP的port的相关信息，包括port的name、network_id、mac_address、admin_state_up、status、device_id、device_owner信息
tags	array	公网IP的标签
created_at	string	创建资源时的 UTC 日期和时间戳
updated_at	string	最近一次更新资源时的 UTC 日期和时间戳
revision_number	integer	公网IP的版本
project_id	string	公网IP所属的项目ID

## 请求示例

```
GET http://{endpoint}/v2.0/floatingips
```

## 正常响应示例

```
{
  "floatingips": [{
    "id": "7e1d4f82-b14e-43d3-b4ad-6d9751487293",
    "tenant_id": "bbd16f62d534430bbdf8ca96bf02a3",
    "floating_ip_address": "172.16.10.227",
    "floating_network_id": "0d4df3da-de1e-4a5e-9606-9178efd4a42c",
    "router_id": null,
    "port_id": null,
    "fixed_ip_address": null,
    "status": "DOWN",
    "description": "",
    "qos_policy_id": null,
    "port_details": null,
    "tags": [],
    "created_at": "2022-03-30T03:42:04Z",
    "updated_at": "2022-03-30T03:42:04Z",
    "revision_number": 0,
    "project_id": "bbd16f62d534430bbdf8ca96bf02a3"
  }]
}
```

## 正常响应代码

200

## 错误码

401

## 创建公网IP

### 功能介绍

创建一个公网IP

### URI

POST /v2.0/floatingips

## 请求消息

参数	参数类型	是否必选	Description
floatingips	object	公网IP对象	
floating_network_id	boolean	是	公网IP地址申请自外部网络的ID
qos_policy_id	string	否	公网IP绑定的qos策略
description	string	否	公网IP的描述
port_id	string	否	绑定公网IP的port的ID
floating_ip_address	string	否	公网IP的地址
fixed_ip_address	string	否	绑定公网IP的port的IP地址
project_id	string	否	公网IP所属的项目ID。

## 响应消息

参数	参数类型	描述
floatingips	object	公网IP对象
id	string	公网IP的ID
tenant_id	string	公网IP所属的用户ID
floating_ip_address	string	公网IP的地址
floating_network_id	string	公网IP地址申请自外部网络的ID
router_id	string	绑定公网IP所属router的ID
port_id	string	绑定公网IP的port的ID

参数	参数类型	描述
fixed_ip_addresses	string	绑定公网IP的port的IP地址
status	string	公网IP的运行状态
description	string	公网IP的描述
qos_policy_id	string	公网IP绑定的qos策略
port_details	string	绑定公网IP的port的相关信息, 包括port的name、network_id、mac_address、admin_state_up、status、device_id、device_owner信息
tags	array	公网IP的标签
created_at	string	创建资源时的 UTC 日期和时间戳
updated_at	string	最近一次更新资源时的 UTC 日期和时间戳
revision_number	integer	公网IP的版本
project_id	string	公网IP所属的项目ID

## 请求示例

```
POST http://{endpoint}/v2.0/floatingips
```

```
{
  "floatingip": {
    "floating_network_id": "0d4df3da-de1e-4a5e-9606-9178efd4a42c"
  }
}
```

## 正常响应示例



```
{
  "floatingip": {
    "id": "7e1d4f82-b14e-43d3-b4ad-6d9751487293",
    "tenant_id": "bbd16f62d534430bbdf8ca96bf02a3",
    "floating_ip_address": "172.16.10.227",
    "floating_network_id": "0d4df3da-de1e-4a5e-9606-9178efd4a42c",
    "router_id": null,
    "port_id": null,
    "fixed_ip_address": null,
    "status": "DOWN",
    "project_id": "bbd16f62d534430bbdf8ca96bf02a3",
    "description": "",
    "qos_policy_id": null,
    "port_details": null,
    "tags": [],
    "created_at": "2022-03-30T03:42:04Z",
    "updated_at": "2022-03-30T03:42:04Z",
    "revision_number": 0
  }
}
```

## 正常响应代码

201

## 错误码

400, 401, 404, 409

## 获取公网IP

### 功能介绍

获取一个公网IP的详情

### URI

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

参数	是否必选	描述
floatingip_id	是	待查询的公网IP的ID

## 请求消息

## 响应消息

参数	参数类型	描述
floatingips	object	公网IP对象
id	string	公网IP的ID
tenant_id	string	公网IP所属的用户ID
floating_ip_address	string	公网IP的地址
floating_network_id	string	公网IP地址申请自外部网络的ID
router_id	string	绑定公网IP所属router的ID
port_id	string	绑定公网IP的port的ID
fixed_ip_addresses	string	绑定公网IP的port的IP地址
status	string	公网IP的运行状态
description	string	公网IP的描述
qos_policy_id	string	公网IP绑定的qos策略
port_details	string	绑定公网IP的port的相关信息，包括port的name、network_id、mac_address、admin_state_up、status、device_id、device_owner信息
tags	array	公网IP的标签

参数	参数类型	描述
created_at	string	创建资源时的 UTC 日期和时间戳
updated_at	string	最近一次更新资源时的 UTC 日期和时间戳
revision_number	integer	公网IP的版本
project_id	string	公网IP所属的项目ID

### 请求示例

```
GET http://{endpoint}/v2.0/floatingips/145c964a-1951-4a34-b62f-441e0d3e2675
```

### 正常响应示例

```
{
  "floatingip": {
    "id": "7e1d4f82-b14e-43d3-b4ad-6d9751487293",
    "tenant_id": "bbd16f62d534430bbdf8ca96bf02a3",
    "floating_ip_address": "172.16.10.227",
    "floating_network_id": "0d4df3da-de1e-4a5e-9606-9178efd4a42c",
    "router_id": null,
    "port_id": null,
    "fixed_ip_address": null,
    "status": "DOWN",
    "description": "",
    "qos_policy_id": null,
    "port_details": null,
    "tags": [],
    "created_at": "2022-03-30T03:42:04Z",
    "updated_at": "2022-03-30T03:42:04Z",
    "revision_number": 0,
    "project_id": "bbd16f62d534430bbdf8ca96bf02a3"
  }
}
```

## 正常响应代码

200

## 错误码

401, 403, 404

## 更新公网IP

### 功能介绍

更新一个公网IP

### URI

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

参数	是否必选	描述
floatingip_id	是	待查询的公网IP的ID

### 请求消息

参数	参数类型	是否必选	Description
floatingips	object	公网IP对象	
port_id	string	否	绑定公网IP的port的ID
fixed_ip_address	string	否	绑定公网IP的port的IP地址
qos_policy_id	string	否	公网IP绑定的qos策略
description	string	否	公网IP的描述

### 响应消息

参数	参数类型	描述
floatingips	object	公网IP对象
id	string	公网IP的ID
tenant_id	string	公网IP所属的用户ID
floating_ip_address	string	公网IP的地址
floating_network_id	string	公网IP地址申请自外部网络的ID
router_id	string	绑定公网IP所属router的ID
port_id	string	绑定公网IP的port的ID
fixed_ip_addresses	string	绑定公网IP的port的IP地址
status	string	公网IP的运行状态
description	string	公网IP的描述
qos_policy_id	string	公网IP绑定的qos策略
port_details	string	绑定公网IP的port的相关信息，包括port的name、network_id、mac_address、admin_state_up、status、device_id、device_owner信息
tags	array	公网IP的标签
created_at	string	创建资源时的 UTC 日期和时间戳
updated_at	string	最近一次更新资源时的 UTC 日期和时间戳
revision_number	integer	公网IP的版本
project_id	string	公网IP所属的项目ID

## 请求示例

```
PUT http://{endpoint}/v2.0/floatingips/145c964a-1951-4a34-b62f-441e0d3e2675
```

```
{  
  "floatingip": {  
    "port_id": null  
  }  
}
```

## 正常响应示例

```
{  
  "floatingip": {  
    "id": "a3fdad57-8f24-41d3-95cf-1f94c7bc04b1",  
    "tenant_id": "45d6353dc853402a8b8987feaccb7429",  
    "floating_ip_address": "172.16.10.130",  
    "floating_network_id": "0d4df3da-de1e-4a5e-9606-9178efd4a42c",  
    "router_id": null,  
    "port_id": null,  
    "fixed_ip_address": null,  
    "status": "ACTIVE",  
    "description": "",  
    "qos_policy_id": "87d450e9-72b9-4711-ac8f-6b0a111e2223",  
    "port_details": null,  
    "tags": [],  
    "created_at": "2022-03-30T06:14:42Z",  
    "updated_at": "2022-03-30T06:18:17Z",  
    "revision_number": 4,  
    "project_id": "45d6353dc853402a8b8987feaccb7429"  
  }  
}
```

## 正常响应代码

200

## 错误码

400, 401, 404, 409, 412

## 删除公网IP

### 功能介绍

删除一个公网IP

### URI

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

参数	是否必选	描述
floatingip_id	是	待查询的公网IP的ID

### 请求消息

### 请求示例

```
DELETE http://{endpoint}/v2.0/floatingips/145c964a-1951-4a34-b62f-441e0d3e2675
```

### 正常响应代码

204

### 错误码

401, 404, 412

## 获取公网IP池列表

### 功能介绍

获取公网IP池列表

### URI

```
GET /v2.0/floatingip_pools
```

## 响应消息

参数	参数类型	描述
floatingip_pools	array	公网IP池列表
subnet_id	string	子网的ID
tenant_id	string	公网IP池所属的项目ID
network_id	string	网络名称
subnet_name	string	子网名称
cidr	string	子网的CIDR
project_id	string	公网IP池所属的项目ID

## 请求示例

```
GET http://{endpoint}/v2.0/floatingip_pools
```

## 正常响应示例

```
{
  "floatingip_pools": [
    {
      "subnet_id": "cdec285c-b157-48aa-900c-e77f6bd958e5",
      "tenant_id": "26a7980765d0414dbc1fc1f88cdb7e6e",
      "network_id": "db193ab3-96e3-4cb3-8fc5-05f4296d0324",
      "subnet_name": "public-subnet",
      "cidr": "192.16.10.0/24",
      "project_id": "26a7980765d0414dbc1fc1f88cdb7e6e"
    }
  ]
}
```

## 正常响应代码



200

## 错误码

401

## 7.10 发布记录

### 01 <2022-05-31>

变更类型	变更说明
更新	<ul style="list-style-type: none"><li>* <a href="#">列举虚拟网卡</a>: 在响应消息列表中, 新增qos_network_policy_id字段。</li><li>* <a href="#">列举QoS policy</a>: 在响应消息列表中, 新增is_default字段。</li><li>* <a href="#">给QoS policy设置带宽</a>: 在请求消息列表中, 新增bandwidth_limit_roule:direction字段。</li><li>* <a href="#">给QoS policy设置带宽</a>: 在响应消息列表中, 新增direction字段。</li><li>* <a href="#">更新路由器</a>: 在请求消息列表中, 移除portforwardings字段。</li><li>* <a href="#">列举路由器</a>: 在响应消息列表中, 移除portforwardings、flavor_id、ha字段。</li><li>* <a href="#">创建安全组</a>: 在请求消息列表中, 新增stateful字段。</li><li>* <a href="#">更新安全组</a>: 在请求消息列表中, 新增stateful字段。</li><li>* <a href="#">列举安全组</a>: 在响应消息列表中, 新增stateful字段。</li><li>* <a href="#">列举安全组规则</a>: 在响应消息列表中, 新增normalized_cidr字段。</li><li>* <a href="#">列举公网IP</a>: 在响应消息列表中, 新增port_details字段。</li></ul>

**咨询热线：400-100-3070**

北京易捷思达科技发展有限公司：

北京市海淀区西北旺东路10号院东区1号楼1层107-2号

南京易捷思达软件科技有限公司：

江苏省南京市雨花台区软件大道168号润和创智中心4栋109-110

邮箱：

[contact@easystack.cn](mailto:contact@easystack.cn) (业务咨询)

[partners@easystack.cn](mailto:partners@easystack.cn)(合作伙伴咨询)

[marketing@easystack.cn](mailto:marketing@easystack.cn) (市场合作)