## 7.1 Neutron网络服务概述

### 7.1.1 什么是Neutron？

neutron服务是OpenStack的核心服务，源自早期的compute-network服务。Neutron 允许 OpenStack 用户创建和管理网络对象，如网络、子网、端口和路由等，而这些网络对象正是其他 OpenStack 服务正常运行所需的网络资源。

### 7.1.2 Neutron的概念

1. API Server：API Server 是 Neutron 的控制器，用户对网络资源的请求全部先交由 API Server 处理。 API Server 接到用户的请求后，会调用后端插件进行具体的任务实现。

1. 网络插件Plugin和代理agent：插件与代理的主要功能在于实现由 API Server 转发的网络资源请求，如网络端口的插拔 、 路由增删 、 网络和子网创建以及提供 IP 地址等 。
2. Flat网络：在 Flat 网络中，不同计算节点上的全部实例接人同一个大二层网  
   络内，全部实例共享此网络，不存在 VLAN 标记或其他网络隔离技术 。 接人 Flat 网络的全部实例通过数据中心核心路由接人 Internet 。最简单的网络类型。Flat网络实现隔离要通过多网卡实现。
3. VLAN网络:VLAN 网络允许用户使用外部物理网络中的 VLAN ID 创建多个租户或provider网络 。VLAN 网络通过 VLAN ID 进行二层网络的隔离，相同 VLAN ID 内部的实例可以实现自由通信，而不同 VLAN 之间的实例通信则需要经过三层路由的转发。 主要用于私有云环境。
4. VxLAN和GRE网络：

GRE 和 VxLAN 是一种网络封装协议，基于这类封装协议可以创建重叠（ overlay ）网络以实现和控制不同计算节点实例之间的通信 。 GRE 和 VxLAN 的主要区别在于 GRE 网络通过 IP 包进行数据传输，而 VxLAN 通过 UDP 包进行数据传输， GRE 或 VxLAN 数据包在流出节点之前会被打上相应的 GRE 或 VxLAN 网络 ID (Segmentation ID ），而在进入节点后对应的 ID 会被剥离，之后再进入节点内部的虚拟网络进行数据转发 。

在 GRE 和 VxLAN 网络中，路由器的主要作用在于通过实例浮动 IP 提供外部网络对实例的直接访问。主要用于公有云环境中。

1. 端口：在 OpenStack 网络中是一种虚拟接口设备，用于模拟物理网络接口 。 在Neutron 中端口是网络设备连接到某个虚拟网络的接人点 。
2. 子网：子网（ Subnet）代表的是一个 IP 段和相关的配置状态 。
3. 子网池：约束子网所能使用的 IP。
4. 路由：在 Neutron 中，路由 （ Router）是个用以在不同网络中进行数据包转发的逻辑组件，即路由是个虚拟设备，在特定插件支持下，路由还提供了 L3 和 NAT 功能，以使得外部网络（不一定是 Internet）与租户私有网络之间实现相互通信。
5. 安全组：安全组 （ Security Group ）是虚拟防火墙的规则集合，这些防火墙规则对外部访问实例和实例访问外部的数据包实现了端口级别（ Port Level ） 的控制 。 安全组使用默认拒绝策略( Default Deny Policy ），其仅包含允许特定数据流通过的规则 。
6. 网络东西和南北流向：

* 租户内部子网之间的数据访问通常称为东西向通信。
* 位于租户内网中的应用要对外提供服务，则必须实现外部网络与租户网络彼此之间的通信，这包括了外网访问内网和内网访问外网两种方式，通常内外网之间的访问称为南北向通信。

1. SNAT源地址转换：虚机访问外部网络需要进行SNAT。
2. DNAT目的地址转换：外部网络访问虚拟需要进行DNAT。
3. 网络命名空间：在 Linux 系统中，网络命名空间（ Network Namespace ）就是一个虚拟的网络设备， 网络命名空间有独立的路由表、 iptables 策略和接口设备等，网络命名空间彼此之间完全隔离。

### 7.1.3 Neutron架构

在 Open Stack 中，网络服务使用 Neutron-server 进程提供服务 API 接口，用户通过 Neutron-server 提供的 API可以进行网络插件的管理配置。 通常为了实现网络配置数据的永久性存储，网络插件需要访问 OpenStack 的中心数据库 。 Neutron 项目的架构如图 7-1 所示，其中 Neutron-server 不仅向云用户提供网络服务API，还向计算服务 Nova 和控制面板服务 Horizon 提供网络资源请求 API ，同时 Neutron-server 与 Neutron 插件需要访问 Keystone 服务以进行身份验证，而为了实现彼此交互， Neutron 的服务代理与插件需要访问消息队列服务 。

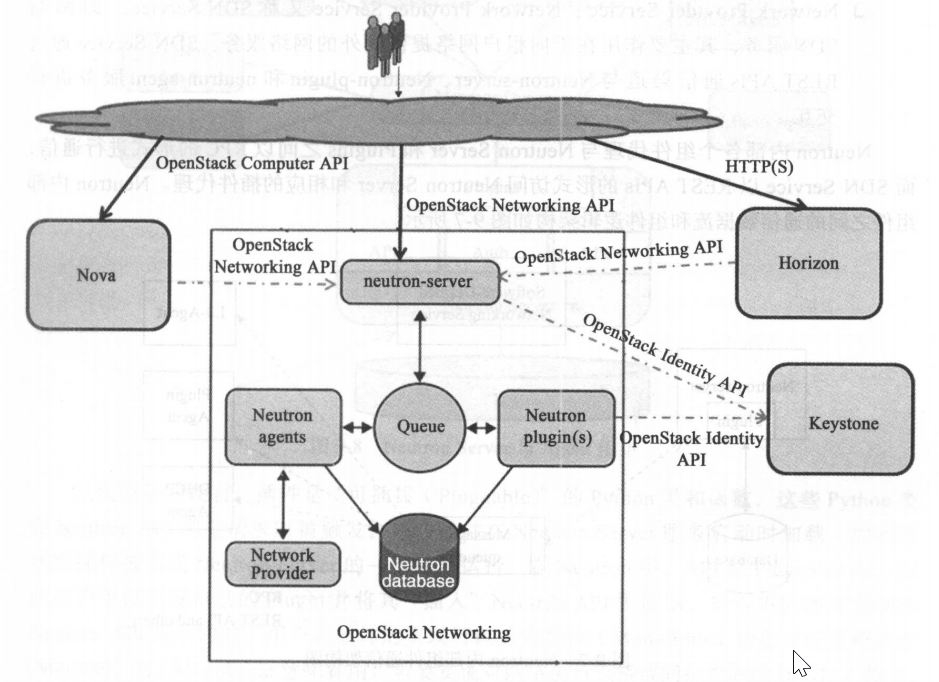


图7-1 Neutron服务架构

Neutron 是一个由分布式组件构成的网络服务，其内部组件包括 Neutron Server、插件代理（ Plugin Agent ）、 DHCP 代理 、 L3 代理和计量代理（ Metering Agent）等其他 SDN 服务、关于 Neutron 服务组件的功能和作用描述如下 ：

* Neutron server：包括neutron-server 服务进程和 neutron-\*-plugin 服务，或者说Server API和neutron插件构成了neutron-server。

1. Server API（端口是9696）：接收和响应外部的网络管理请求。API 接收请求后负责调用相应的 Plugin 进行请求处理 。
2. neutron plugin：Plugin 又负责调用Plugin Agent 进行最终的任务处理并维护 OpenStack 网络逻辑状态，因此 Plugin 是Neutron 中主要的数据库访问者 。

* Plugin Agent(neutron-\*-agent) ：插件代理是最终的网络提供者（ Network Provider），即租户网络服务请求在经过 Neutron Server 和 Plugin 之后，最终通过 Plugin Agent 来实现租户网络请求任务 。插件代理服务需要访问消息队列以便和插件服务进行交互，同时插件代理需要访问数据库以便存储网络变更信息 。 因此插件代理也是 Neutron 中主要的数据库访问者 。

1. neutron-linuxbridge-agent： 负责创建桥接网卡，让虚拟机可以上网，桥接到不同的网络上。
2. neutron-metadata-agent： 配置nova-metadata-api来实现虚拟机的定制化（如开机执行脚本，设置hostname）

* DHCP Agent : DHCP Agent 服务（ neutron-dhcp-agent）主要为租户网络提供 DHCP  
  服务,自动分配IP地址。

DHCP Agent 对于 Neutron 支持的全部网络 Plugins 都是相同的，即 DHCP 代理并不依赖用户选用的 Plugin 。 DHCP Agent 服务也需要访问消息队列以便同网络Plugins 进行交互。

* L3 Agent : L3 Agent 在 Provider 类型的网络中并不是必须的，而在 Self-Service 网  
  络中却是必须的 。 L3 Agent 服务主要在外网访问租户网络中的虚拟机时提供 L3  
  Route 功能 。 L3 Agent 也需要访问消息队列 。

* 高级服务：

LBaas(load balance as a service):负载均衡，mitaka不能实现。

fwaas

vpnaas...

openstack创建的虚拟机和Host在同一个网络，即为flat网络。

### 7.1.4 Neutron Plugin

OpenStack中虚拟机实例需要二层网络，还可能要实现路由、防火墙、VPN和负载均衡等高级网络功能。这些网络服务和技术可以通过软件与硬件设备结合的方式来实现，在具体的技术和产品选型过程中，又有各种开源软件网络实现技术和不同的厂商设备可供选择。为了可以实现灵活多样的OpenStack网络，Neutron以Plugin的形式来实现各种网络技术。Neutron包含了实现各种网络技术的插件。

1. ML2插件

ML2插件解耦了对不同网络驱动的调用，将驱动分为两个部分：

* TypeDirver：代表不同的网络隔离类型,如Flat，Local，VLAN，GRE和VxLAN。TypeDirver负责维护特定类型网络所需的状态信息，执行Provider网络验证和Self-Service网络分配工作。
* MechanismDriver: 主要负责提取TypeDriver所建立的信息，并确保将正确应用到用户启用的特定网络机制中。如Linux Bridge和Open vSwitch等。Linux Bridge和Open vSwitch都支持Flat，VLAN，GRE和VxLAN这4种TypeDriver。

Neutron的插件类型：

* Core插件：负责L2网络通信和IP管理，如网络、子网池和端口的创建和删除。
* Service插件：负责提供三层、四层、七层的高级网络服务，如L3路由、防火墙、VPN和负载均衡。

ML2是最典型的Core插件。H版之后，Neutron中已经集成了ML2插件，并统一使用ML2插件与不同的插件代理协调工作以实现各种L2层的网络技术。

### 7.1.5 Neutron网络类型

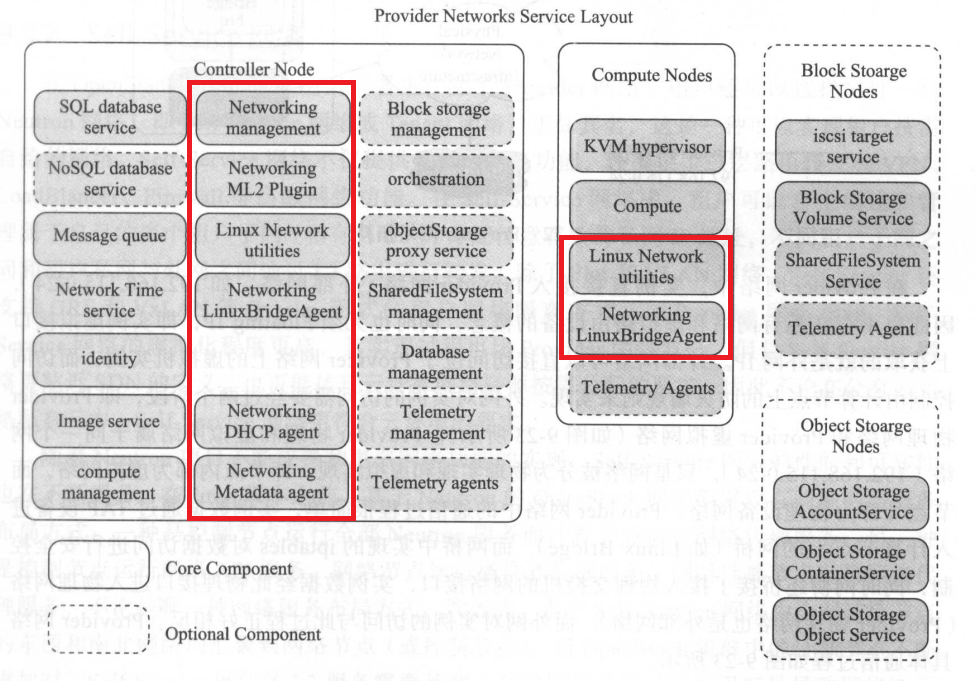
在 Neutron 网络的具体实施过程中，用户可以选择两种网络类型， 即 Provider网络  
和Self-Service网络，后者又称 Tenant 网络或 Project 网络。二者主要区别在于 Provider  
网络只能由云管理员创建，并且实例可接入Provider网络（又称外部网络或物理网络），  
Provider 网络中不存在租户私网，也没有L3路由和Floating IP地址，同时Provider 网络  
仅支持 Flat和VLAN网络类型； Self-Service 网络是对 Provider的扩展 ，在租户可以创建Self-Service 网络之前必须先由云管理员创建 Provider 网络， Self-Service网络具有L3 服务 ，因此实例可以接人租户私网，除了云管理员，其他非授权用户也可以自助管理和使用 Self-Service 网络 ， 括用以将私网与外网进行互联的Router 。 在Self-Service网络中 ，外部网络（如 Internet）对接人私网的实例访问通过 Floating IP 来实现 。

1. provider网络

特点：提供二层网络，三层路由和更改成的VPN服务，LoadBalaner服务，firewall服务由物理网络设备来实现。

* 服务部署：

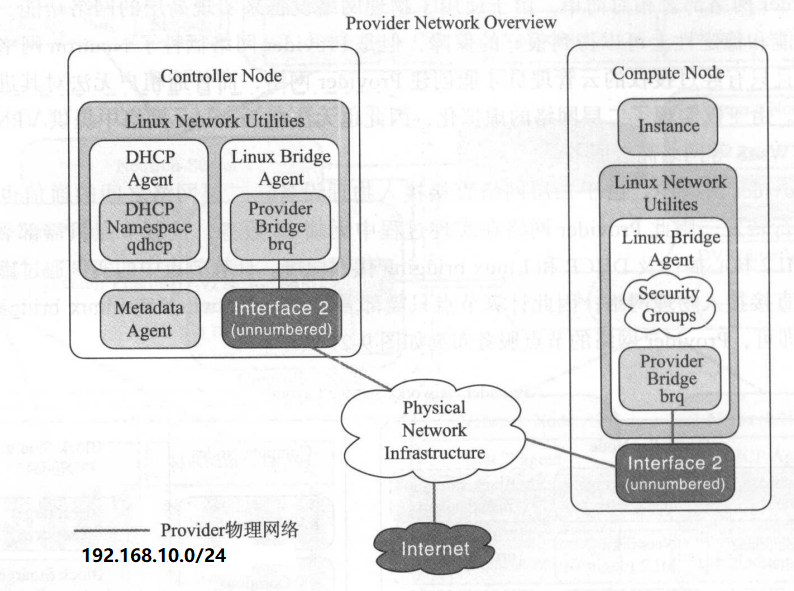
控制节点只需部署 API-Server 、 ML2 核心插件及 DHCP 和 Linux bridging 代理。 计算节点中的实例通过虚拟二层交换机直接接人物理网络，因此计算节点只需部署如 Open vSwitch 或 Linux bridging 等代理软件。Provider网络服务布局图如下：



Provider网络服务布局图

* 网络拓扑

Provider 网络拓扑架构只需在控制节点和计算节点中分别规划一块网卡，并将其接入网络即可 。 如果采用的是 VLAN 网络，则将交换机接口配置为 Trunk 模式，  
节点只需一块网卡便可通过多个 VLAN ID 来实现不同网络的隔离 。 如果采用的是 Flat  
网络，由于 Flat 网络没有 Tagging ，如要配置多个 Flat 网络则需要节点提供同样数量的网卡 。 Provider 网络的拓扑架构如下：

 Provider网络拓扑架构

在 Provider 网络中，实例直接接人 Provider 网络（外部网络，如 192.168.10.0/24 ),  
因此也不存在私有网络和虚拟路由设备的概念，同时也无须 Floating IP ，即实例虚拟接口  
上获取的就是外网 IP 。 外部网络可以直接访问位于 Provider 网络上的虚拟机实例，而访问控制由计算节点上的防火墙规则来实现 。

1. Self-Service网络

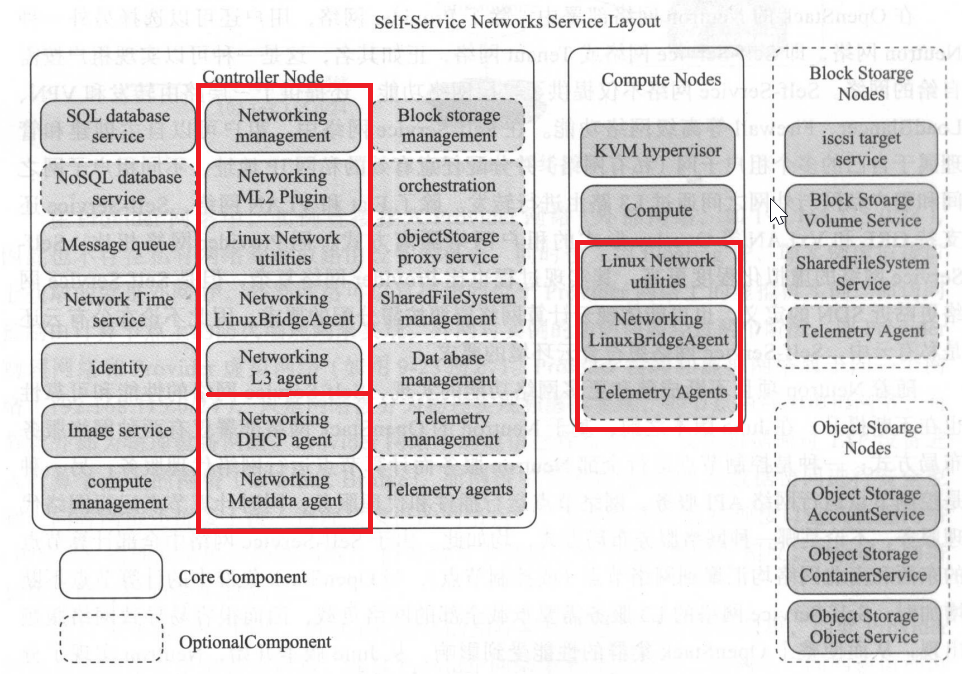
特点：

1. 可以实现租户按需自给的网络。
2. Self-Service 网络不仅提供了二层网络功能，还提供了三层路由转发和VPN 、  
   LoadBalancer 、Firewall 等高级网络功能。
3. 在 Self-Service 网络中，租户可以自主创建和管理属于自己的多个租户子网（私有网络）并分配任意有效的私网 IP 地址，不同租户子网之间和租户私网与外网之间通过 L3 路由进行转发。
4. 除了 Flat 和 VLAN 网络， Self-Service 还支持 GRE 和 VxLAN等 Overlay 形式的租户网络隔离方式 。
5. 同 Provider 网络相比， Self-Service 网络的虚拟化程度更高，其实现过程也比 Provider 网络复杂，但是 Self-Service 网络更贴近 SDN 的定义，也更能体现云计算网络资源按需分配的要求，因此不论在公有云还是私有云中， Self-Service 网络更符合云环境的要求 。

* 部署

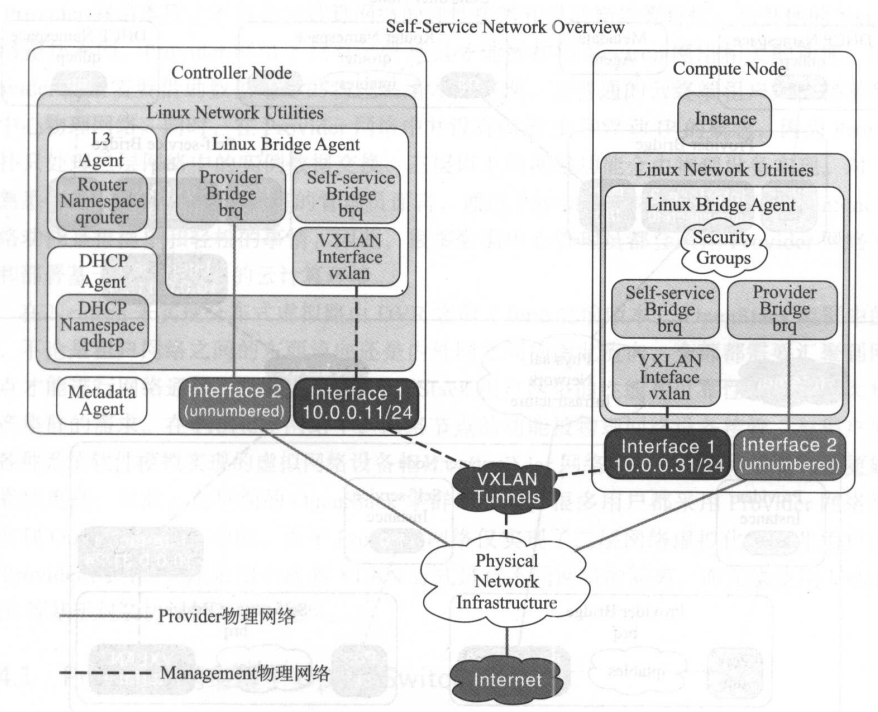
Self-Service 网络的 L3 层功能采用虚拟技术来实现，因此 Self-Service 网络在实现过程中需要运行更多的网络服务组件。由于部署 Self-Service 网络之前，必须首先实现 Provider 网络，因此 Self-Service网络的服务布局相对于 Provider 主要是多了L3 Agent .

部署服务结构图如下：



Self-Service网络部署结构图

在租户可以创建 Self-Service 网络之前，管理员必须事先创建 Provider 网络， 然后租户创建的 Self-Service 网络通过 NAT 形式接入 Provider 物理网络 。 计算节点上的实例创建时，接入 Self-Service 网络的实例可以直接访问外部网络（如 Internet ） 。 但是，外部网络要访问计算节点实例，则必须通过实例 Floating IP ，并通过 DNAT 才可实现 。 Self-Service 网络通常使用 GRE 或 VxLAN 这类 Overlay 网络，即将虚拟网络进行封装后叠加（ Overlay)到物理网络上进行传输 。 由于 Self-Service 本身是租户内部私有网络（虚拟网络）， Self-Service 网络中不同节点之间要实现互通，需要借助隧道（ Tunnel ）封装技术来实现（如GRE 或 VxLAN ） ，通常节点之间的 Tunnel 建立在物理管理网络之上（Management Physical Network ), Self-Service 网络拓扑如下。



Self-Service网络拓扑图

Provider 网络仅支持 Flat 和 VLAN 网络，而 Self-Service 网络中可以并存Provider 网络，因此 Self-Service 网络支持 Flat、VLAN 、GRE 和VxLAN 网络类型，并且 Self-Service 网络中的实例也可以直接接入 Provider 物理网络，并使用 Flat 或 VLAN 方式进行网络隔离。