目 录

Openstack 之 neutron 服务

周威光整理

2017-06-22

1 neutron 简介

neutron 是虚拟化网络的一种实现方式，为什么要网络虚拟化，主要有两个方面的需求，一是互联网行业数据中心的基本特征就是服务器的规模偏大。进入云计算时代后，其业务特征变得更加复杂，包括：虚拟化支持、多业务承载、资源灵活调度等。与此同时，互联网云计算的规模不但没有缩减，反而更加庞大。这就给云计算的网络带来了巨大的压力。二是数据中心（Data Center）中的物理网络是固定的、需要手工配置的、单一的、没有多租户隔离的网络。而云架构往往是多租户架构，这意味着多个客户会共享单一的物理网络。因此，除了提供基本的网络连接能力以外，云还需要提供网络在租户之间的隔离能力；同时云是自服务的，这意味着租户可以通过云提供的 API 来使用虚拟出的网络组建来设计，构建和部署各种他们需要的网络。OpenStack 云也不例外，其通过 Neutron 项目在物理网络环境之上提供满足多租户要求的虚拟网络和服务。

2 neutron 功能介绍

Neutron 作为虚拟化网络的一种实现方式，提供的网络虚拟化能力主要包括如下：

1. 二层到七层网络的虚拟化
2. 租户隔离性
3. 网络安全性
4. 网络高可用性和扩展性
5. 更高级的服务，包括 LBaaS，FWaaS，VPNaaS 等

2.1 二层到七层网络的虚拟化

2.1.1 二层虚拟化

二层提供 network,subnet,port 资源，默认采用开源的 Open vSwitch 作为其虚机交换机，同时还支持使用 Linux bridge

1. 网络（network）是一个隔离的二层网段，类似于物理网络世界中的虚拟 LAN (VLAN)。更具体来讲，它是为创建它的租户而保留的一个广播域，或者被显式配置为共享网段。端口和子网始终被分配给某个特定的网络。

   根据创建网络的用户的权限，Neutron network 可以分为：

   (1). Provider network：管理员创建的和物理网络有直接映射关系的虚拟网络;

   (2). Tenant network：租户普通用户创建的网络，物理网络对创建者透明，其配置由 Neutron 根据管理员在系统中的配置决定;

   根据网络的类型，Neutron network 可以分为：

   (1). local network（本地网络）：一个只允许在本服务器内通信的虚拟网络，不知道跨服务器的通信。主要用于单节点上测试。

   (2). Flat network：基于不使用 VLAN 的物理网络实现的虚拟网络。每个物理网络最多只能实现一个虚拟网络。

   (3). VLAN network（虚拟局域网）：基于物理 VLAN 网络实现的虚拟网络。共享同一个物理网络的多个 VLAN 网络是相互隔离的，甚至可以使用重叠的 IP 地址空间。每个支持 VLAN network 的物理网络可以被视为一个分离的 VLAN trunk，它使用一组独占的 VLAN ID。有效的 VLAN ID 范围是 1 到 4094。

   (4). GRE network （通用路由封装网络）：一个使用 GRE 封装网络包的虚拟网络。GRE 封装的数据包基于 IP 路由表来进行路由，因此 GRE network 不和具体的物理网络绑定。

   (5). VXLAN network（虚拟可扩展网络）：基于 VXLAN 实现的虚拟网络。同 GRE network 一样，VXLAN network 中 IP 包的路由也基于 IP 路由表，也不和具体的物理网络绑定。

2. 子网 (subnet) 是一组 IPv4 或 IPv6 地址以及与其有关联的配置。它是一个地址池，OpenStack 可从中向虚拟机 (VM) 分配 IP 地址。每个子网指定为一个无类别域间路由 (Classless Inter-Domain Routing) 范围，必须与一个网络相关联。除了子网之外，租户还可以指定一个网关、一个域名系统 (DNS) 名称服务器列表，以及一组主机路由。这个子网上的 VM 实例随后会自动继承该配置。

3. 端口 (Port) 代表虚拟网络交换机（logical network switch）上的虚机交换端口 (virtual switch port)。虚机的网卡（VIF - Virtual Interface）会被连接到 port 上。当虚机的 VIF 连接到 Port 后，这个 vNIC 就会拥有 MAC 地址和 IP 地址。Port 的 IP 地址是从 subnet 中分配的。

2.1.2 三层虚拟化

三层虚拟化通过一个 Virtual router 提供不同网段之间的 IP 包路由功能，由 Nuetron L3 agent 负责管理

2.1.3 四层到七层虚拟化:

neutron 还提供提供负载均衡，VPN, 防火墙等四层到七层的虚拟化

1. Neutron LBaas (load-balancer-as-a-service) 扩展 (extension) 提供向在多个 Nova 虚机中运行的应用提供负载均衡的方法。它还提供 API 来快速方便地部署负载均衡器。Neutron 默认以 HAProxy 为负载均衡的 driver，同时也支持 A10 network、netscaler、radware 等作为 driver。

2. Neutron 项目的 VPNaaS 是一种 site-to-site 类型的 VPN 解决方案，通过向用户提供 RESETS API，CLI 和 Horizon GUI 去操作 IPSec 来实现。

3. Neutron 提供一种基于 Neutron L3 Agent 的一种网络四层防火墙虚拟化参考实现 Firewall-as-a-service，简称 FWaas。FWaaS 在租户网络边缘实现的虚拟路由器上通过创建防火墙规则实实现。

2.2 租户隔离性

Neutron 实现了不同层次的租户网络隔离性

1. 租户之间的网络是三层隔离的，连通过 VR 做路由都不行，实在要连通的话，需要走物理网络

2. 一个租户内的不同网络之间二层隔离的，需要通过 VR 做三层连通

3. 一个网络内的不同子网也是二层隔离的，需要通过 VR 做三层连通

2.3 网络安全性

1. Neutron 还提供数据网络与外部网络的隔离性。默认情况下，所有虚机通往外网的流量全部走网络节点上的 L3 agent。在这里，内部的固定 IP 被转化为外部的浮动 IP 地址。这种做法一方面保证了网络包能够回来，另一方面也隐藏了内部的 IP 地址。

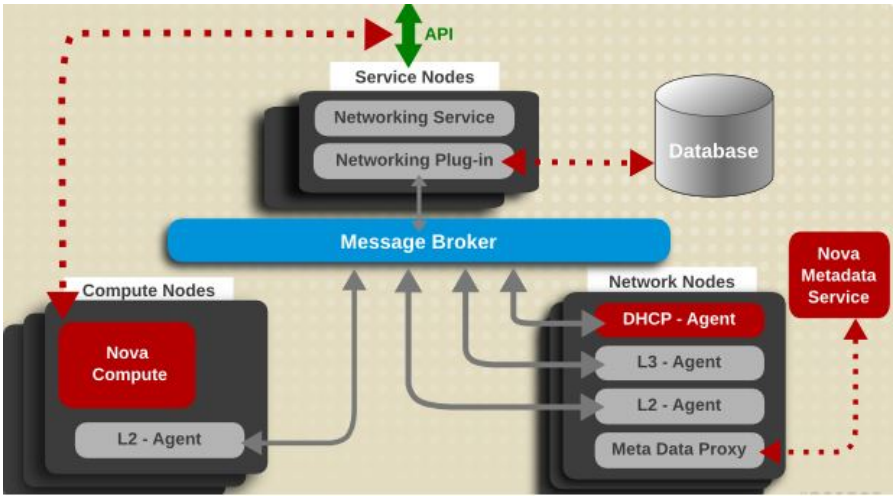
2. Neutron 还是用 Linux iptables 特性，实现其 Security Group 特性，从而保证访问虚机的安全性。

3. Neutron 利用网络控制节点上的 network namespace 中的 iptables，实现了进出租户网络的网络包防火墙，从而保证了进出租户网络的安全性。

2.4 网络高可用性和扩展性

OpenStack 云中可能用于成千上万台虚机，成千上万个租户，因此，Neutron 的数据网络的可用性和扩展性非常重要。Neutron 中，这些特性包括几个层次：

1. 软件架构上，Neutron 实现 OpenStack 标准的去中心化架构和插件机制，有效地保证了其扩展性。如下图所示



2. 支持分布式 Virtual Router(DVR)，默认情况下，L3 agent 部署在网络节点上，这在大规模的云环境中可能会存在性能问题。通过使用 DVR，L3 转发和 NAT 会被分布在计算节点上，这使得计算节点变成了网络节点，这样集中式的网络节点的负载就被分担了。

3. 支持 Virtual Router Redundancy Protocol（VRRP）机制，借助实现 VRRP 协议的软件，来保证 Neutron L3 Agent 的高可用性

4. L2 Population 和 ARP Responder：这两个功能大大减少了网络的复杂性，提交了网络效率，从而促进了扩展性。

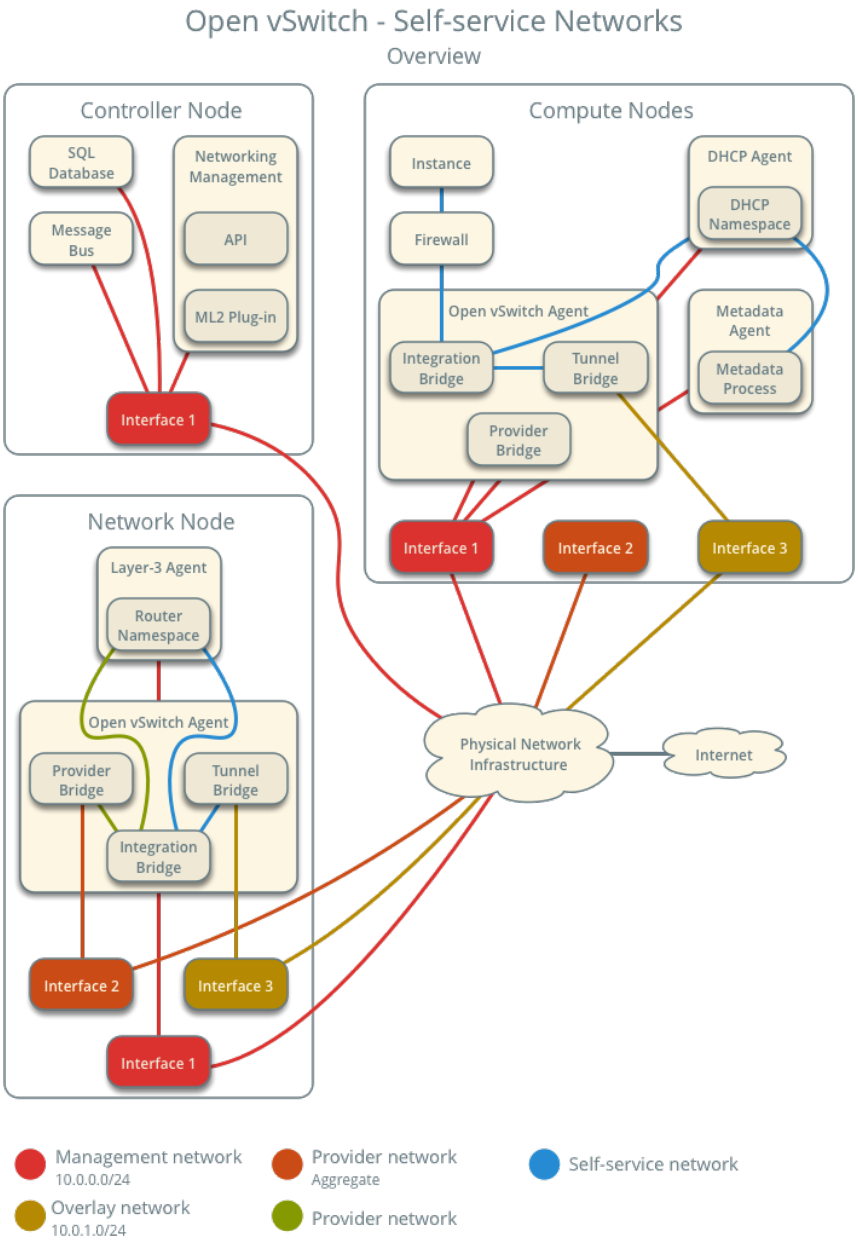
2.5 提供高级服务

在实际的网络中，除了网络的核心功能以外，还有一些普遍应用的网络服务，比如 VPN，Load Balancing 和 Firewall

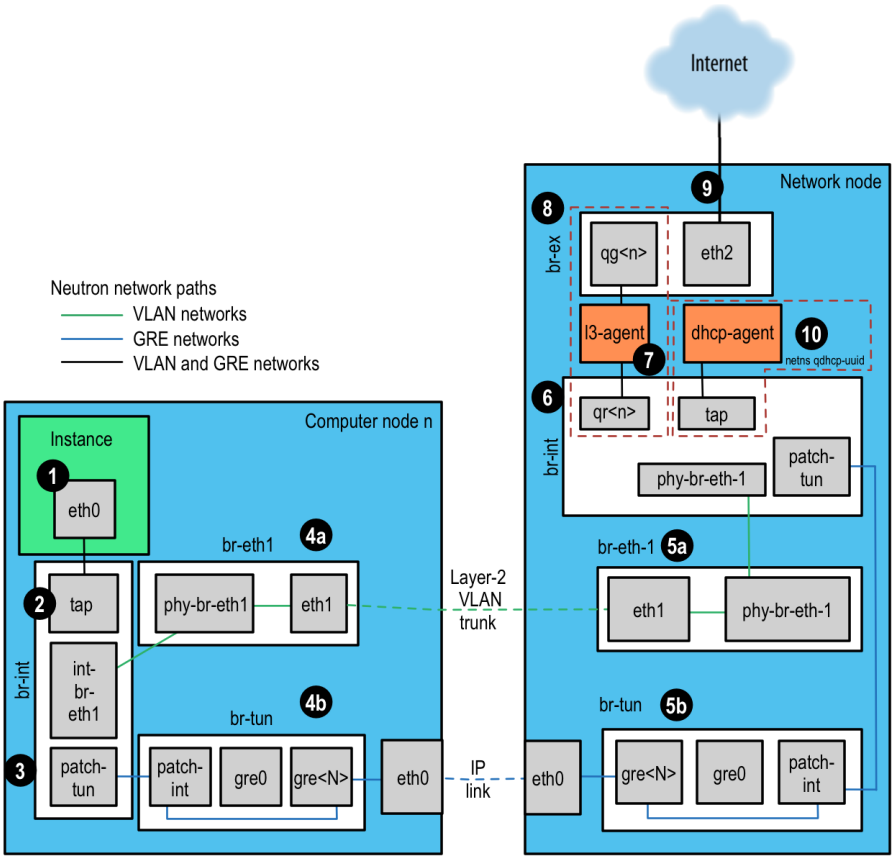
3 neutron 基本架构

neutron 架构包括两个部分：neutron 各服务在机器节点上的分布概况，neutron 各组件及其连接概况

1. neutron 各服务在机器节点上的分布概况，如下图所示



2. neutron 各组件及其连接概况，如下图所示


Internet
Network node
9
8
br-ex
qg<n>
eth2
l3-agent
7
dhcp-agent
10
netns qdhcp-uuid
6
qr<n>
tap
br-int
patch-tun
phy-br-eth-1
br-eth-1
5a
eth1
phy-br-eth-1
br-tun
5b
gre<N>
gre0
patch-int
eth0
IP link
Layer-2 VLAN trunk
Neutron network paths
VLAN networks
GRE networks
VLAN and GRE networks
Computer node n
Instance
1
eth0
br-eth1
4a
2
tap
phy-br-eth1
eth1
br-int
int-br-eth1
br-tun
4b
3
patch-tun
patch-int
gre0
gre<N>
eth0