2.4.5 Nova – 计算服务

讲师：汪洋

目录

1 相关说明

Openstack 是由 Rackspace 和 NASA 共同开发的云计算平台

类似于 Amazon EC2 和 S3 的云基础架构服务

Nova 在 Openstack 中提供云计算服务

超过 140 家企业及 18470 为开发者参与开发


redhat
IBM
nebula
GLUSTER
rackspace
the open cloud company
hp
intel
sina 新浪
CISCO
ERICSSON
NEC
vmware
HUAWEI
f5
NASA
MIRANTIS

nova-api service 接收并响应终端用户计算 API 调用。该服务支持 OpenStack 计算 API， Amazon EC2 和特殊的管理特权 API

nova-api-metadata service 接受从实例元数据发来的请求。该服务通常与 nova-network 服务在安装多主机模式下运行

nova-compute service 一个守护进程，通过虚拟化层API接口创建和终止虚拟机实例。例如： XenAPI for XenServer/XCP， libvirt for KVM or QEMU，VMwareAPI for Vmware

nova-scheduler service 从队列中获取虚拟机实例请求，并确认由哪台计算服务运行该虚拟机

nova-conductor module 协调 nova-compute 服务和 database 之间的交互数据。避免 nova-compute 服务直接访问云数据库。不要将该模块部署在 nova-compute 运行的节点上

nova-network worker daemon 类似于nova-compute服务，接受来自队列的网络任务和操控网络。比如这只网卡桥接或改变iptables规则

nova-consoleauth daemon 在控制台代理提供用户授权令牌

nova-novncproxy daemon 提供了一个通过VNC连接来访问运行的虚拟机实例的代理。支持基于浏览器的novnc客户端

nova-spicehtml5proxy daemon 提供了一个通过 spice 连接来访问运行的虚拟机实例的代理。支持基于浏览器的 HTML5 客户端

nova-xvpnvncproxy daemon 提供了一个通过VNC连接来访问运行的虚拟机实例的代理。支持 OpenStack-specific Java 客户端

nova-cert daemon x509 证书

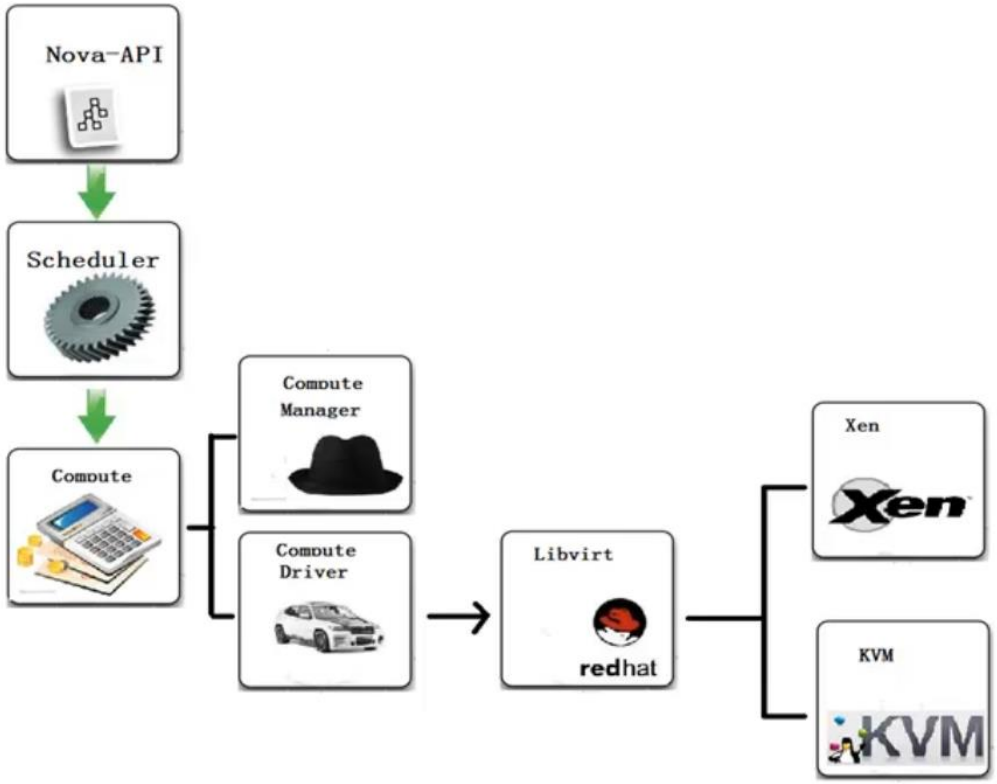
nova-objectstore daemon 一个 Amazon S3 的接口，用于将 Amazon S3 的镜像注册到 OpenStack

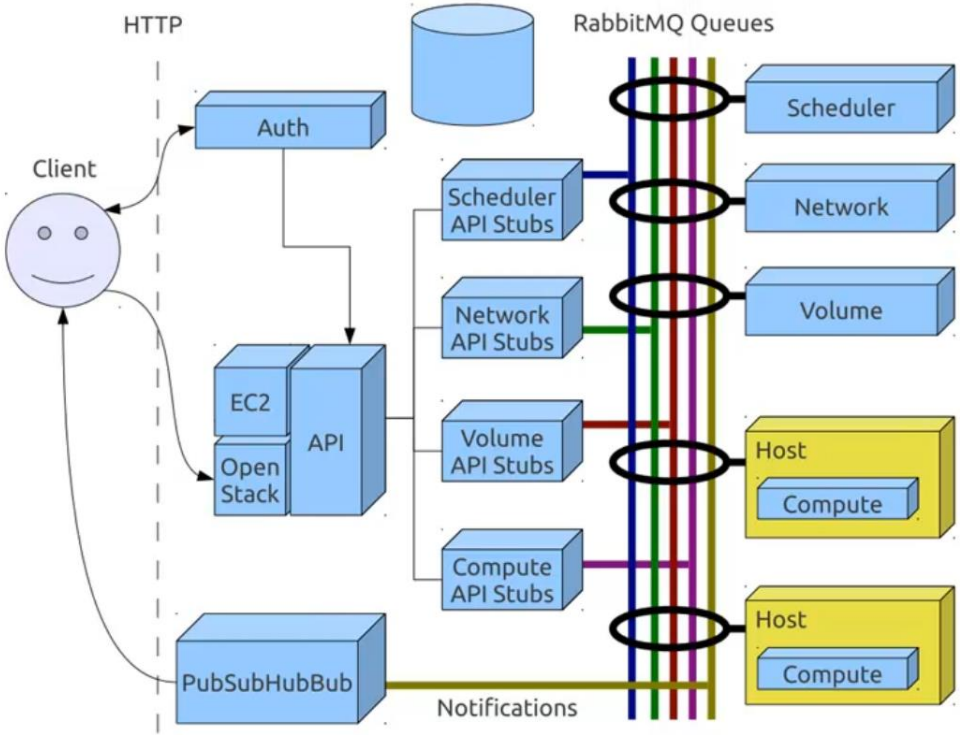
euca2ools client 用于兼容于 Amazon E2 接口的命令行工具

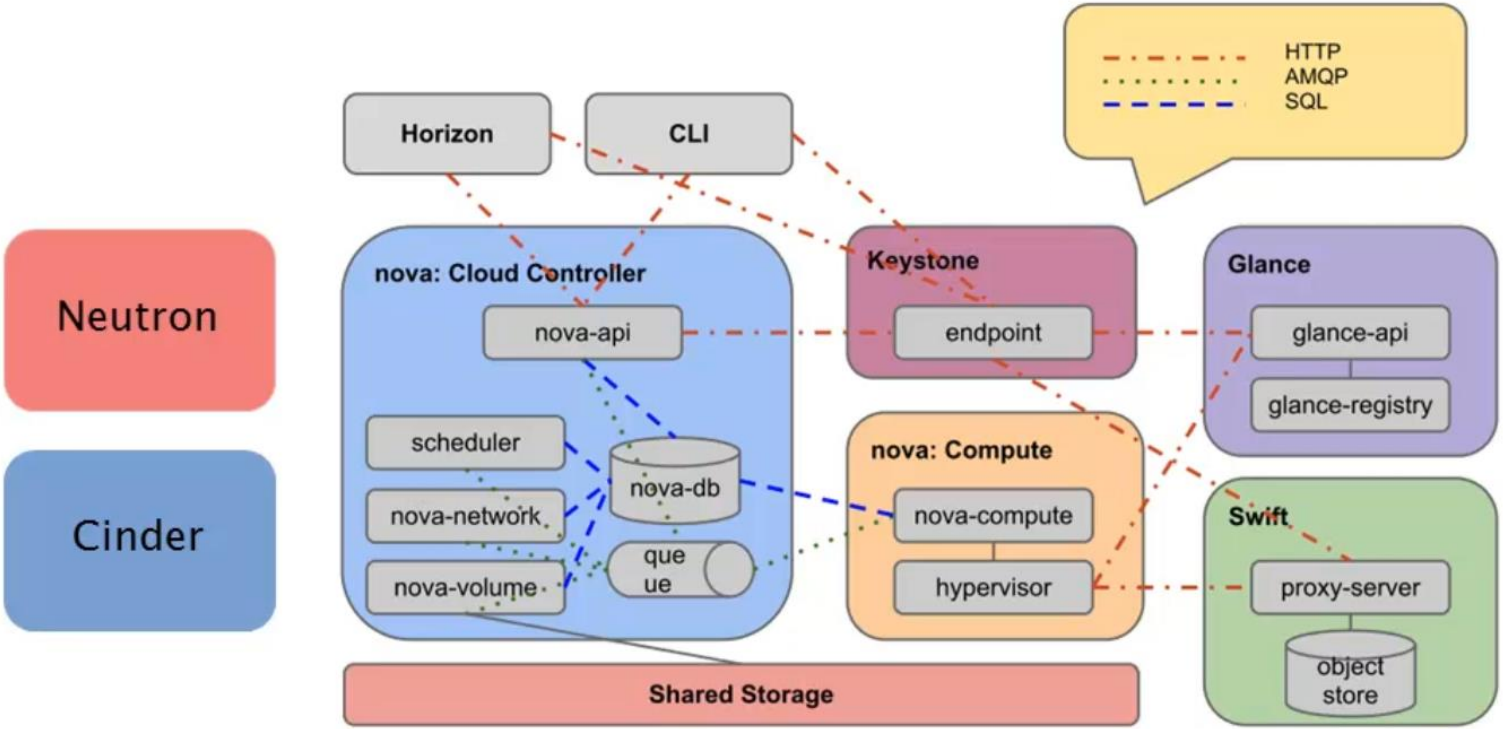
nova client nova命令行工具

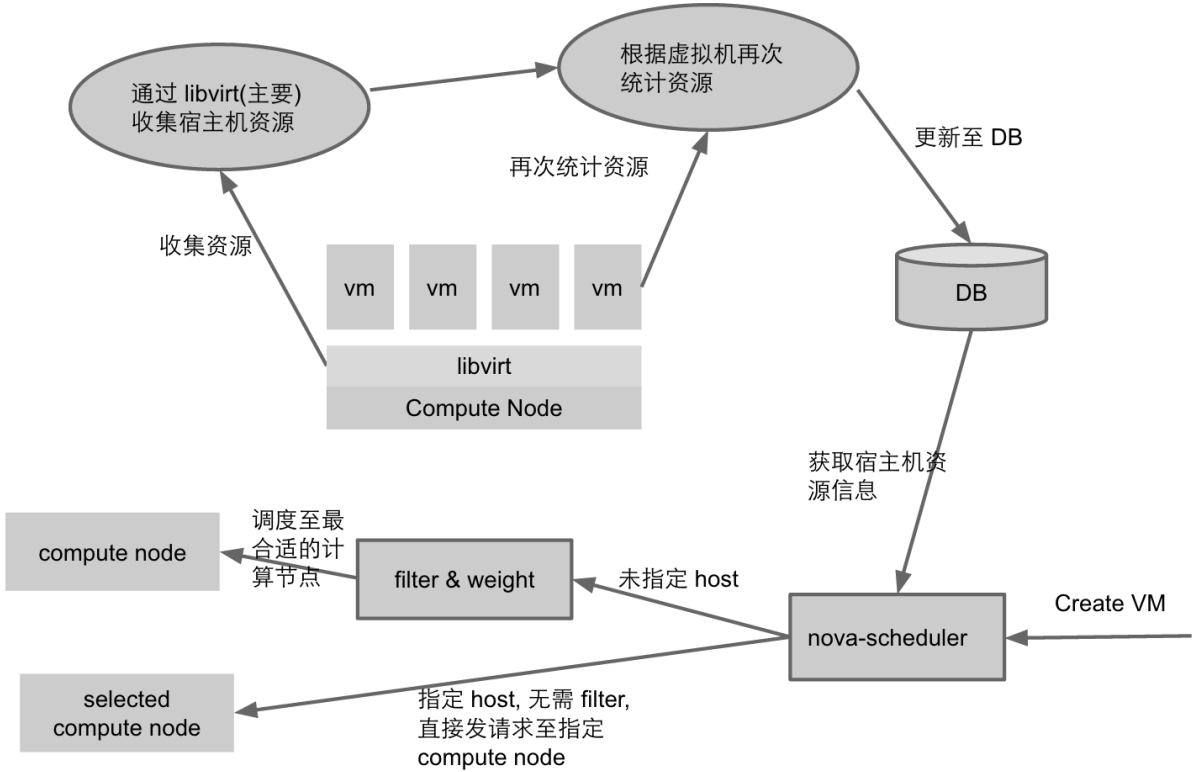
The queue 在进程之间传递消息的中心。通常使用 RabbitMQ

SQL database 保存云基础设置建立和运行时的状态信息


Nova-API
Scheduler
Compute
Compute Manager
Compute Driver
Libvirt
redhat
Xen
KVM


HTTP
RabbitMQ Queues
Auth
Client
Scheduler
Scheduler API Stubs
Network
Network API Stubs
Volume
EC2
API
Open Stack
Volume API Stubs
Host
Compute
Compute API Stubs
Host
Compute
PubSubHubBub
Notifications


HTTP
AMQP
SQL
Horizon
CLI
Neutron
Cinder
nova: Cloud Controller
nova-api
scheduler
nova-db
nova-network
que
ue
nova-volume
Keystone
endpoint
nova: Compute
nova-compute
hypervisor
Glance
glance-api
glance-registry
Swift
proxy-server
object
store
Shared Storage


通过 libvirt(主要)
收集宿主机资源
根据虚拟机再次
统计资源
更新至 DB
再次统计资源
收集资源
vm
vm
vm
vm
DB
libvirt
Compute Node
获取宿主机资
源信息
compute node
调度至最
合适的计
算节点
filter & weight
未指定 host
nova-scheduler
Create VM
selected
compute node
指定 host, 无需 filter,
直接发请求至指定
compute node

2 构建实验

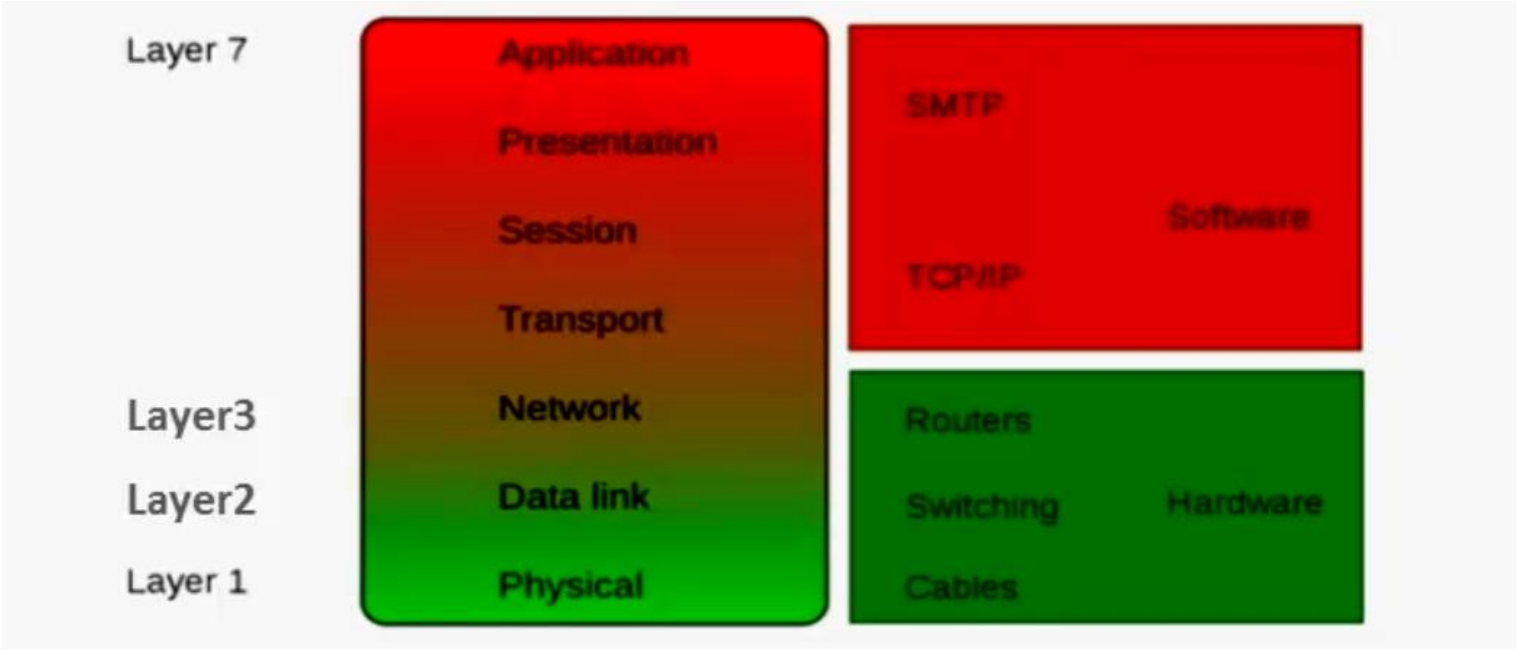
[打开构建文档]

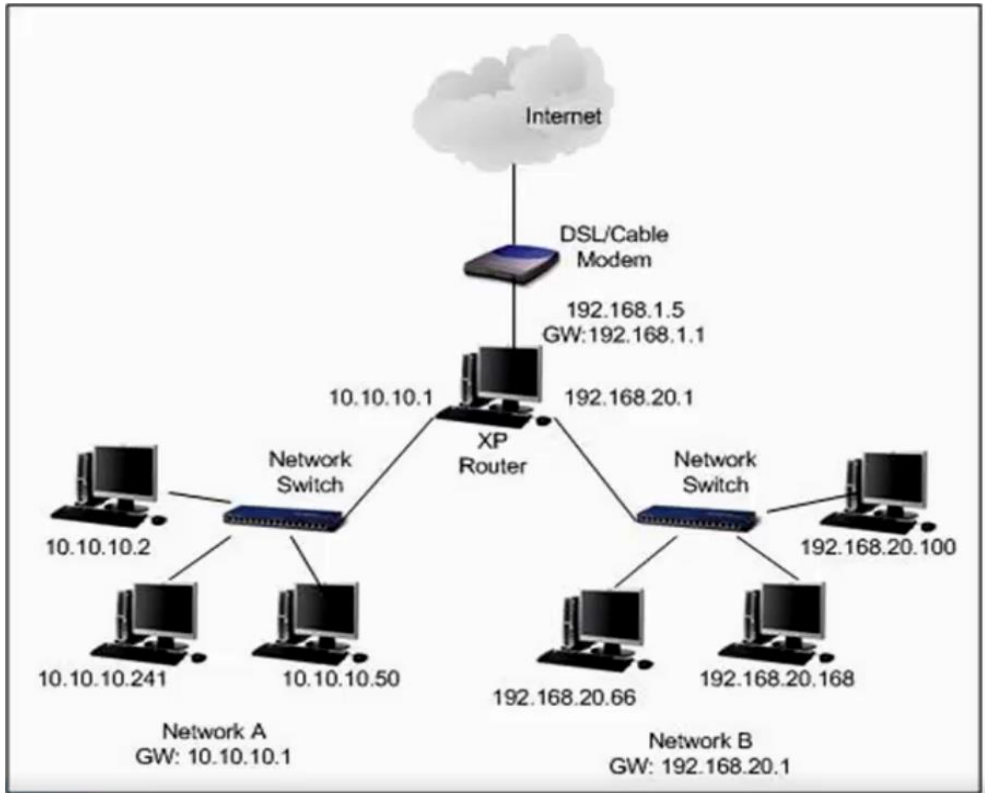
2.4.6 网络基础概念 - Openstack

讲师：汪洋

目录

1 网络相关知识


Layer 7
Application
Presentation
Session
Transport
Network
Data link
Physical
Layer3
Layer2
Layer 1
SMTP
TCP/IP
Software
Routers
Switching
Cables
Hardware


Internet
DSL/Cable
Modem
192.168.1.5
GW:192.168.1.1
10.10.10.1
192.168.20.1
XP
Router
Network
Switch
Network
Switch
10.10.10.2
192.168.20.100
10.10.10.241
10.10.10.50
192.168.20.66
192.168.20.168
Network A
GW: 10.10.10.1
Network B
GW: 192.168.20.1

工作层次不同

- L2/L3

数据转发依据对象不同

- 数据帧（MAC）/数据包（IP）

解决问题不同

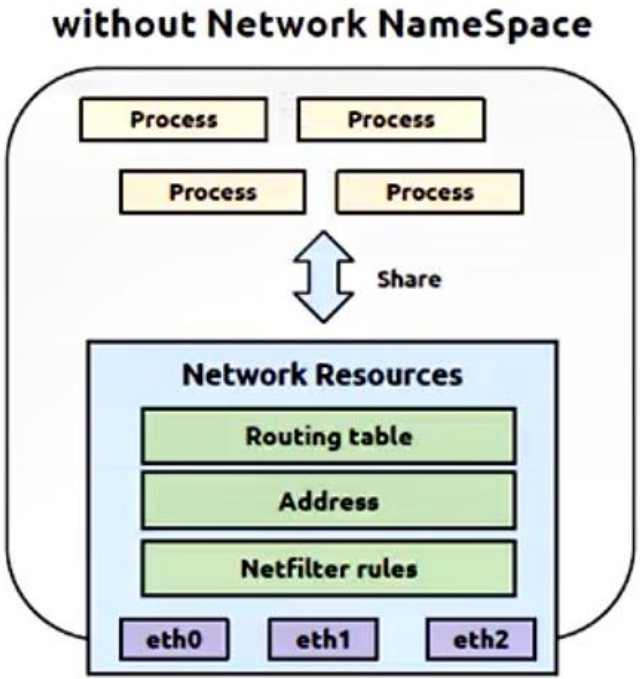
- 同网段互通/不同网段互通

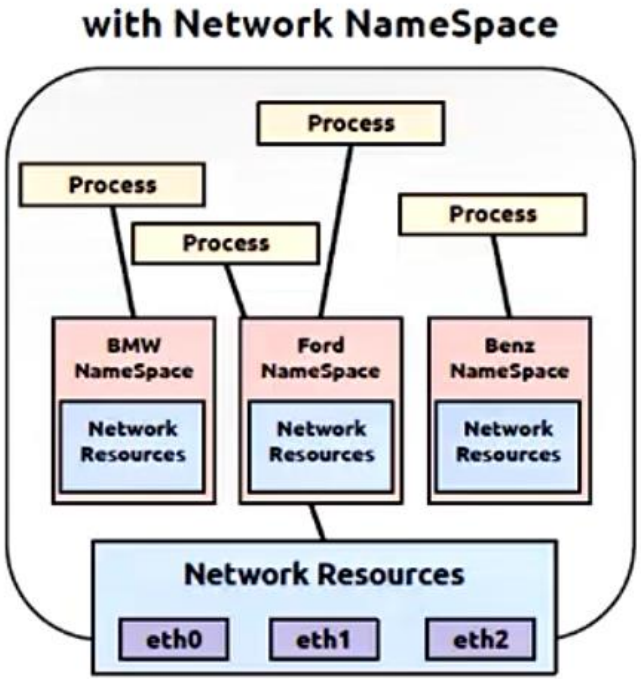
```
[root@network0 ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.20.0.0       *               255.255.255.0   U     0      0        0 eth0
192.168.4.0     *               255.255.255.0   U     0      0        0 eth2
172.16.0.0      *               255.255.255.0   U     0      0        0 br-ex
link-local      *               255.255.0.0     U     1002   0        0 eth0
link-local      *               255.255.0.0     U     1003   0        0 eth1
link-local      *               255.255.0.0     U     1004   0        0 eth2
default         10.20.0.1       0.0.0.0         UG    0      0        0 eth0
```

```
[root@vm1 ~]# ip netns exec router-ns iptables -t nat -nL
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  0.0.0.0/0            192.168.4.51        to:192.168.1.11

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  192.168.1.11         0.0.0.0/0           to:192.168.4.51
SNAT       all  --  192.168.1.0/24       0.0.0.0/0           to:192.168.4.50

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  0.0.0.0/0            192.168.4.51        to:192.168.1.11
```

- 接受所有经过设备的数据包
- 一般用于网络抓包
- Floating IP 功能实现

```
[root@network0 ~]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 52:54:00:C8:67:12
          inet6 addr: fe80::5054:ff:fec8:6712/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:128948 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5935555 (5.6 MiB)  TX bytes:4586 (4.4 KiB)
          Interrupt:10 Base address:0xa000
```


without Network NameSpace
Process
Process
Process
Process
Share
Network Resources
Routing table
Address
Netfilter rules
eth0
eth1
eth2


with Network NameSpace
Process
Process
Process
Process
Process
BMW NameSpace
Ford NameSpace
Benz NameSpace
Network Resources
Network Resources
Network Resources
Network Resources
eth0
eth1
eth2

1、一个数据包（或帧）封装在另一个数据包内；被封装的包转发到隧道端点后再被拆装

2、叠加网络就是使用这种所谓“包内之包”的技术安全的将一个网络隐藏在另一个网络中，然后将网段进行迁移

Vlan

- L2 over L2

GRE

- L3 over L3 （UDP）

Vxlan

- L2 over L3 （UDP）

不用变更底层网络架构重建 L2、L3 通讯

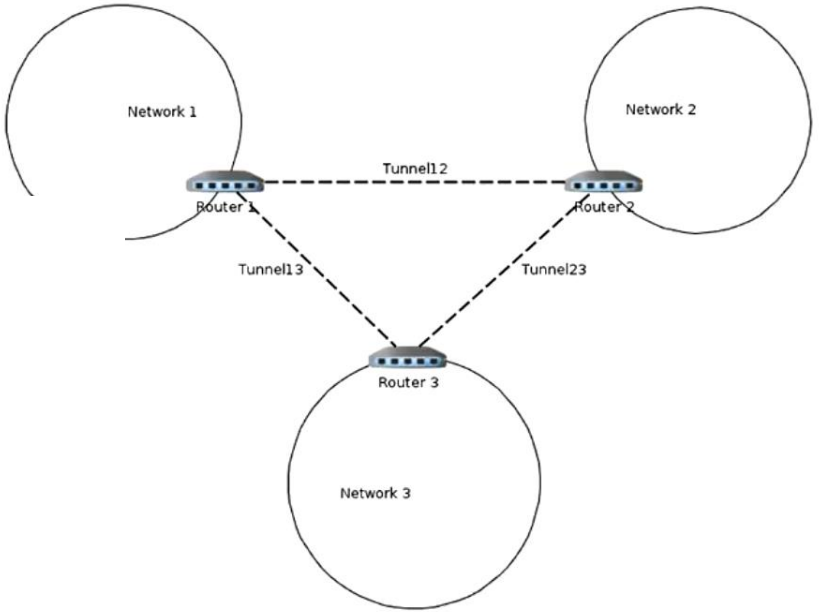
实现不同 Host 之间网络互通

方便迁移

支持网络数量扩大

大规模部署问题

性能问题



数据中心网络数量限制：1 > 4096 > 1600w

物理网络基础设施限制

- 不改变物理网络变更 VM 网络拓扑
- VM 迁移

多租户场景

- 支持 IP 地址重叠

- TAP/TUN
- Bridge
- Physical
- Loopback

网络隔离：Vlan Gre Vxlan

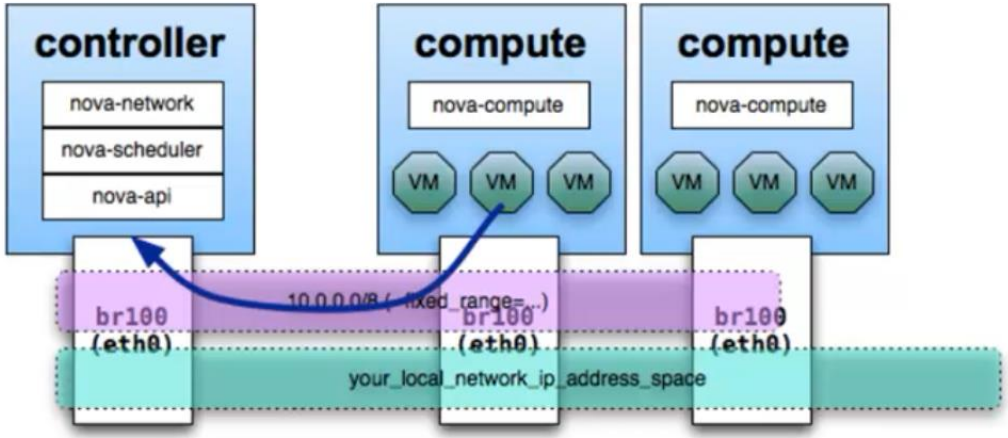
Qos 配置

流量监控

数据包分析

2 Neutron

- Flat
- Flatdhcp
- Vlan



指定一个子网，规定虚拟机能使用的 IP 地址范围

创建实例是，从有效 IP 地址池获取一个 IP，为虚拟机实例分配，然后在虚拟机启动时候注入虚拟机镜像（文件系统）

手动配置好网桥，所有的系统实例都是和同一个网桥连接；网桥连接到网桥的实例组成一个虚拟网络

网络控制器对虚拟机实例进行 NAT 转换，实现与外部的通信

目前配置注入只能对 类 UNIX 操作系统正常工作

网络控制器运行 dnsmasp 作为 DHCP 服务器监听这个网桥

每个用户分配一个 Vlan ，每个用户创建的 网络接口 在同一个 Vlan 中

每个用户分配一个网段，网络控制器上的 DHCP 服务器为所有的 Vlan 分配地址

解决了隔离问题，但是 Vlan 限制为 4096 个，符合私有云

传统桥接模式

用户不能自定义网络

网络隔离

大规模不是

基础需求

高密度

多租户

大规模扩展

虚拟机的可移动性

资源管理自动化

低成本实现

企业级需求

自定义网络

Qos 保证

防火墙

监控、审计

网络连接服务

面向租户 API 接口，用于创建虚拟网络、路由器、负载均衡、关联 网络接口至指定网络和路由

通过 API 接口管理虚拟或物理交换机

提供 Plugin 架构来支持不同的技术平台

Neutron Private Network - 提供固定私网地址

Neutron Public Network - 提供浮动 IP 地址

Network

- 一个 L2 网络单元
- 租户可通过 Neutron API 创建自己的网络

Subnet

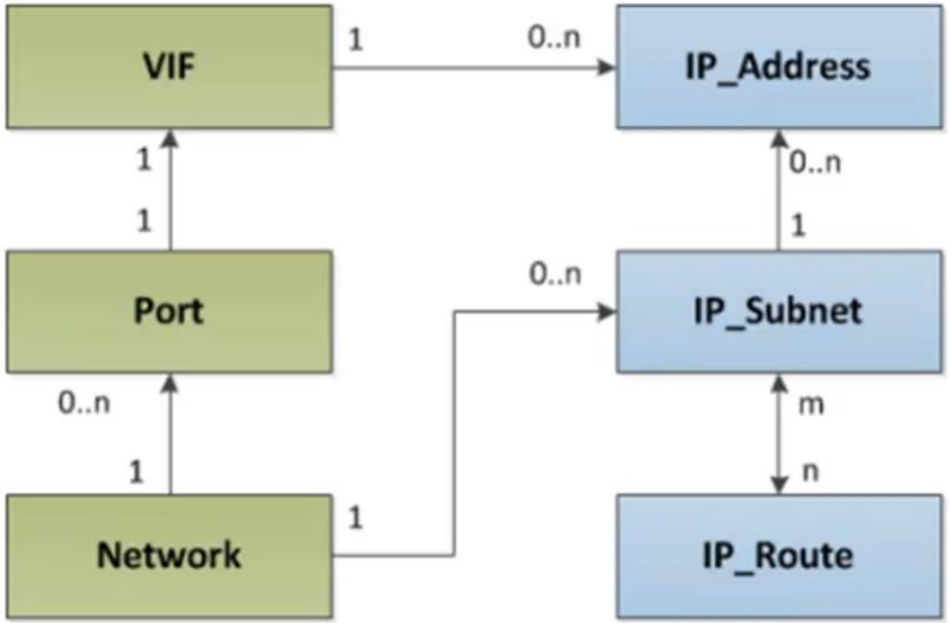
- 一段 IPV4/IPV6 地址段
- 为 Instance 提供私网或公网地址

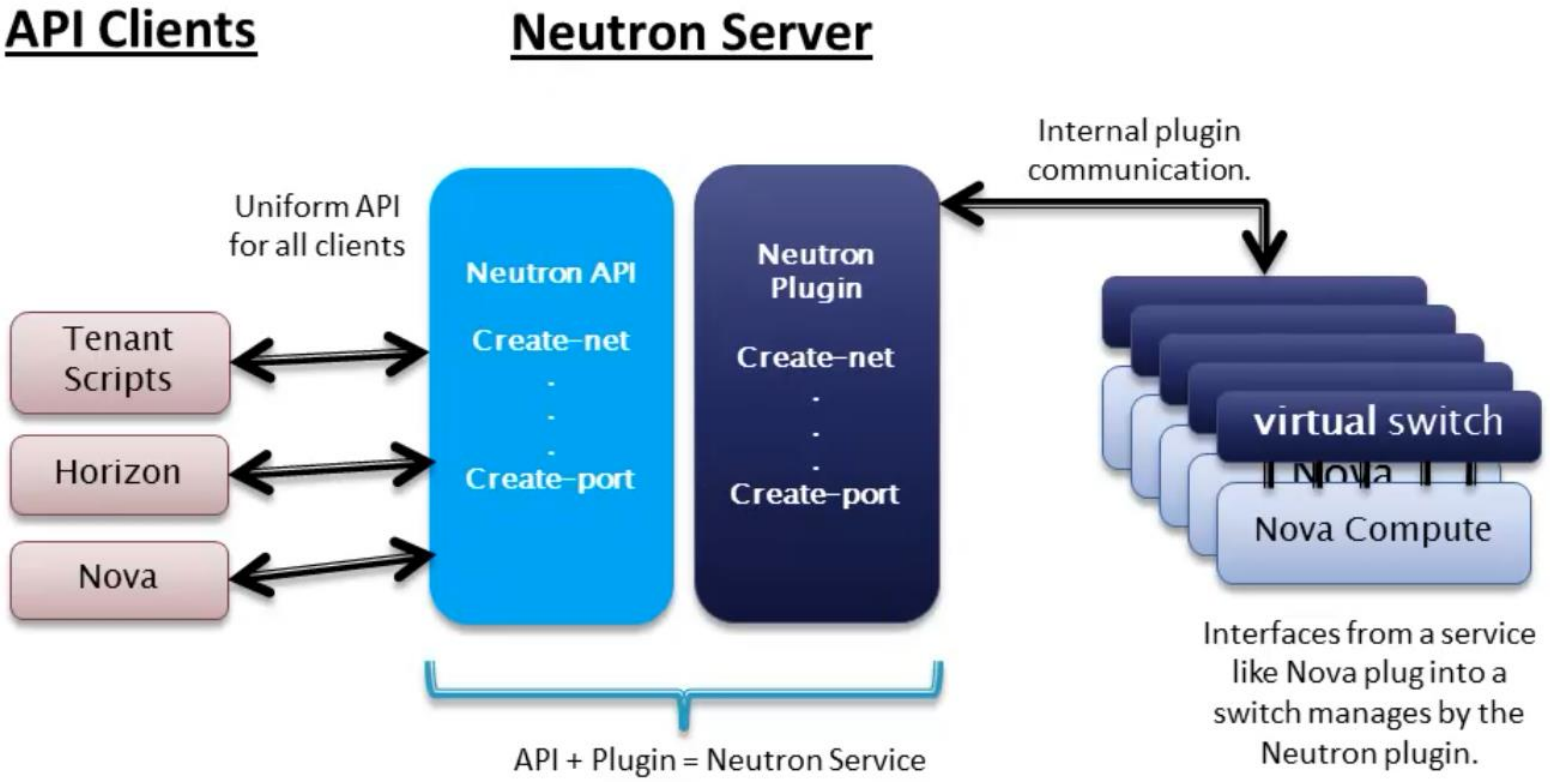
Router

- 三层路由器
- 为租户的 Instance 提供路由功能

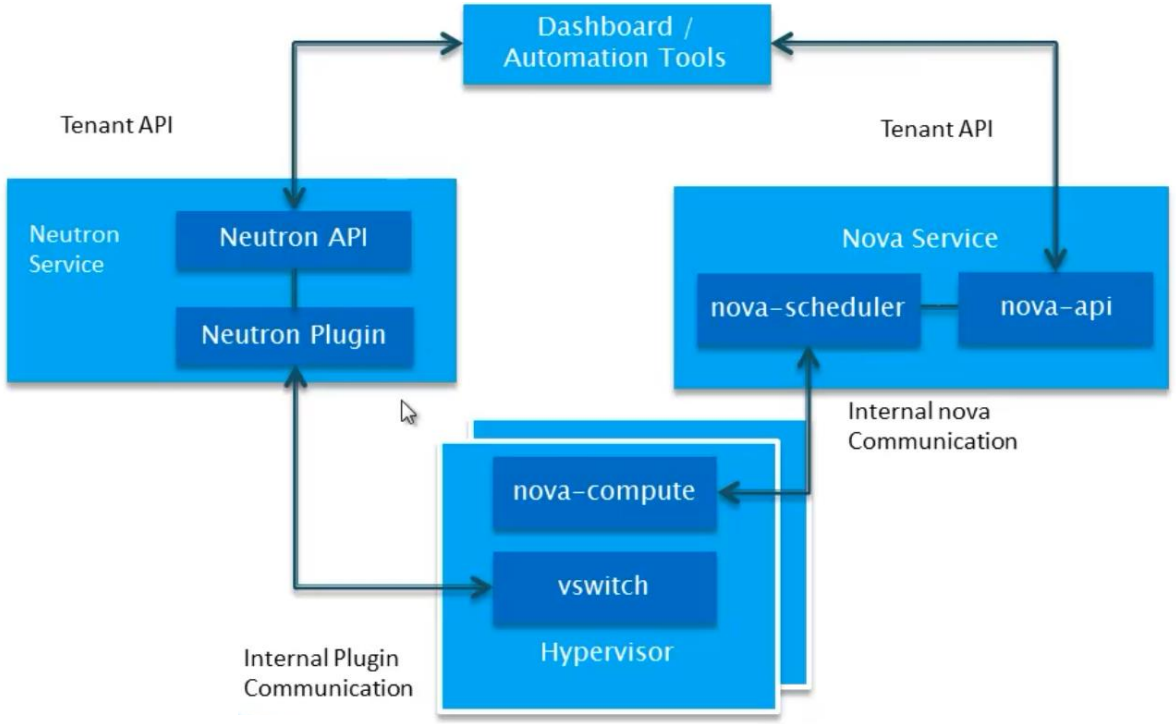
Port

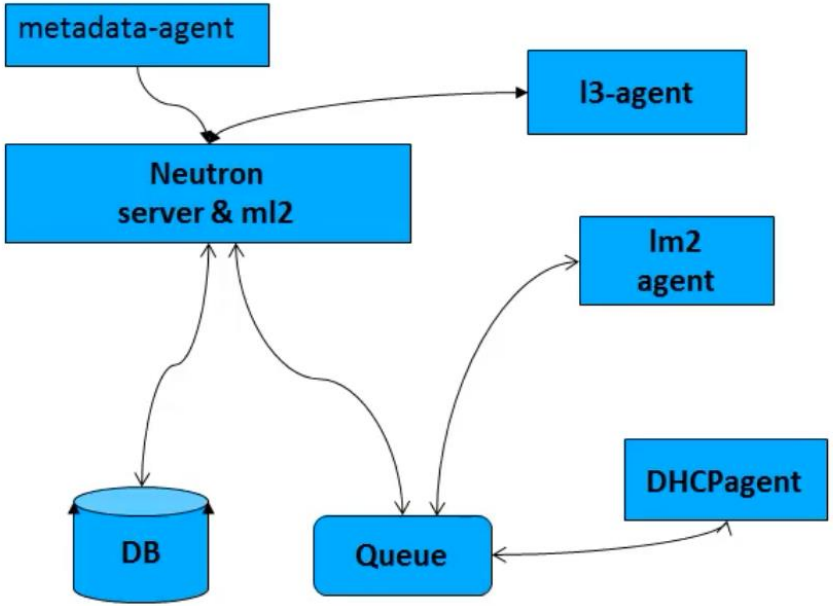
- 虚拟交换机上的端口
- 管理 Instance 的网卡


VIF
1
0..n
IP_Address
1
1
0..n
1
Port
0..n
IP_Subnet
0..n
1
m
n
1
Network
IP_Route


API Clients
Neutron Server
Internal plugin communication.
Uniform API for all clients
Neutron API
Create–net
Create–port
Neutron Plugin
Create–net
Create–port
Tenant Scripts
Horizon
Nova
virtual switch
Nova Compute
Interfaces from a service like Nova plug into a switch manages by the Neutron plugin.
API + Plugin = Neutron Service

- ✓ Open vSwitch
- ✓ Linux Bridge
- ✓ Ciso NX1000
- ✓ Nicira NVP
- ✓ Ryu
- ✓ NEC OpenFlow
- ✓ Floodnight


Dashboard / Automation Tools
Tenant API
Tenant API
Neutron Service
Neutron API
Neutron Plugin
Nova Service
nova-scheduler
nova-api
Internal nova Communication
nova-compute
vswitch
Hypervisor
Internal Plugin Communication



Neutron Server

- 实现 Neutron API 和 API 扩展
- 管理 Network、Subnet、Pod
- 管理 Port 的 IP 地址

ML2 agent

- 运行在每个计算节点上
- 连接虚拟机到网络端口

DHCP agent

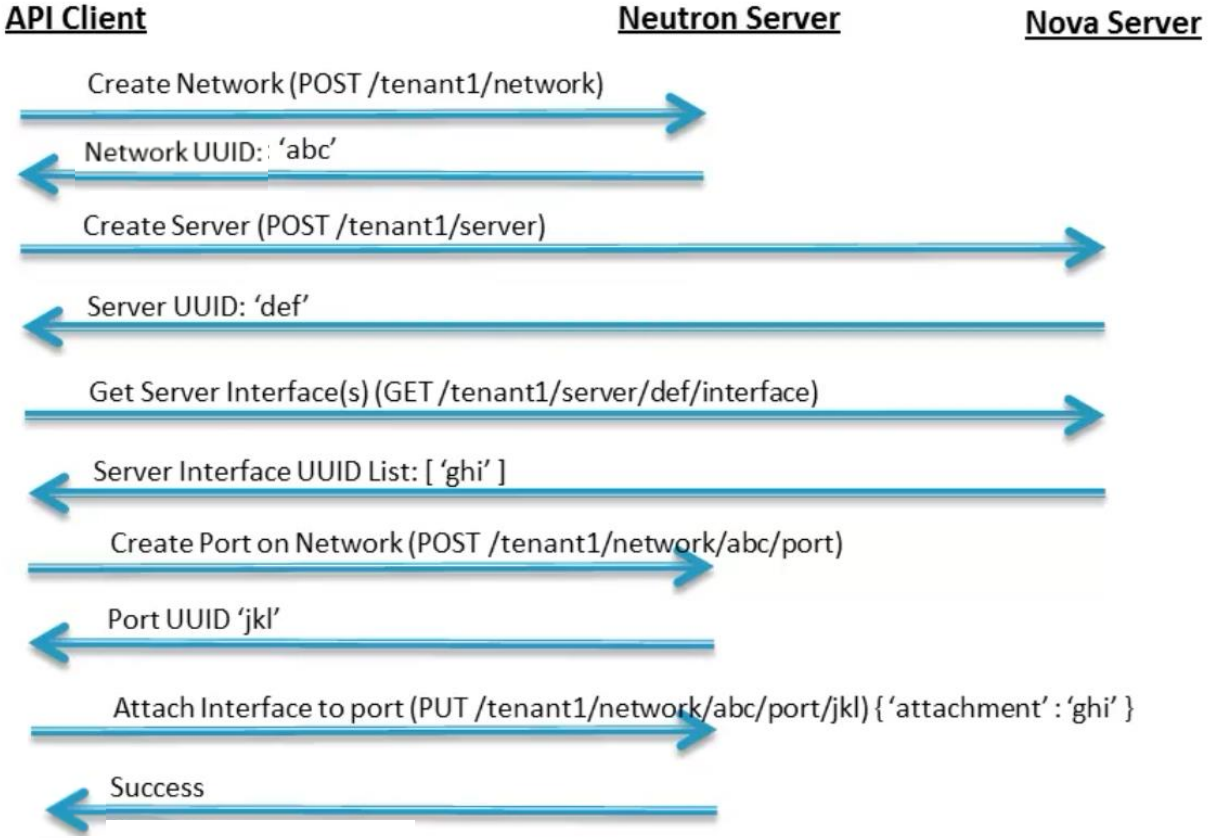
- 负责 DHCP 配置，为虚拟机分配 IP
- 开始/停止 DHCP 服务器以及相关配置

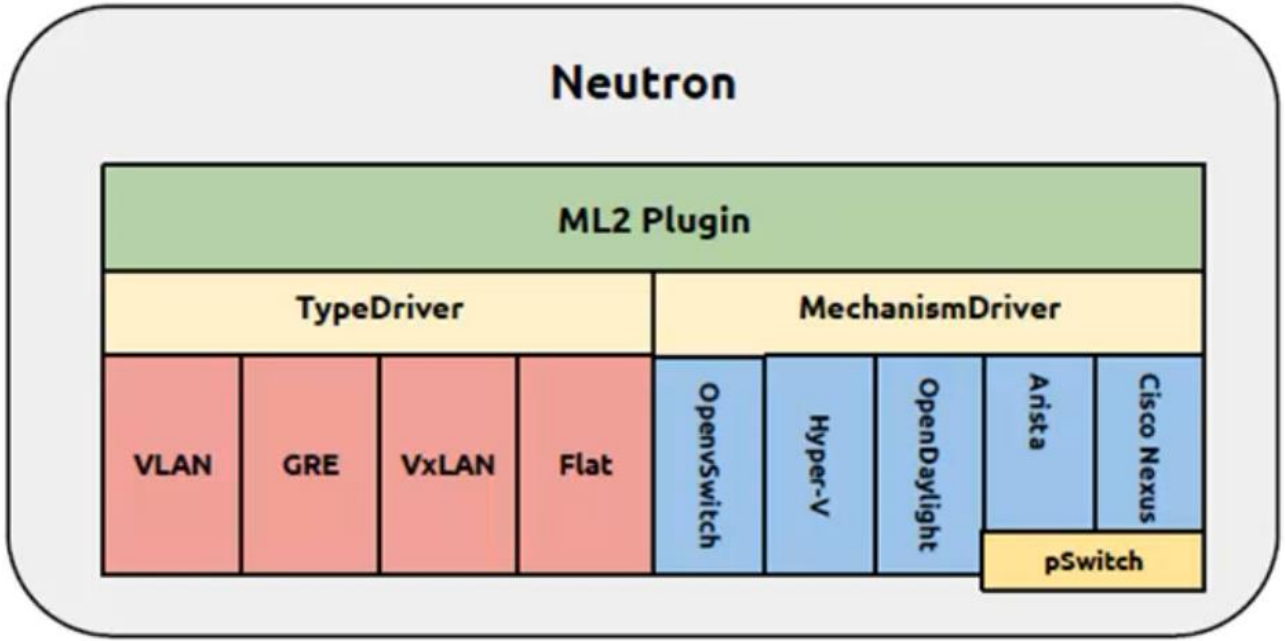
L3-agent

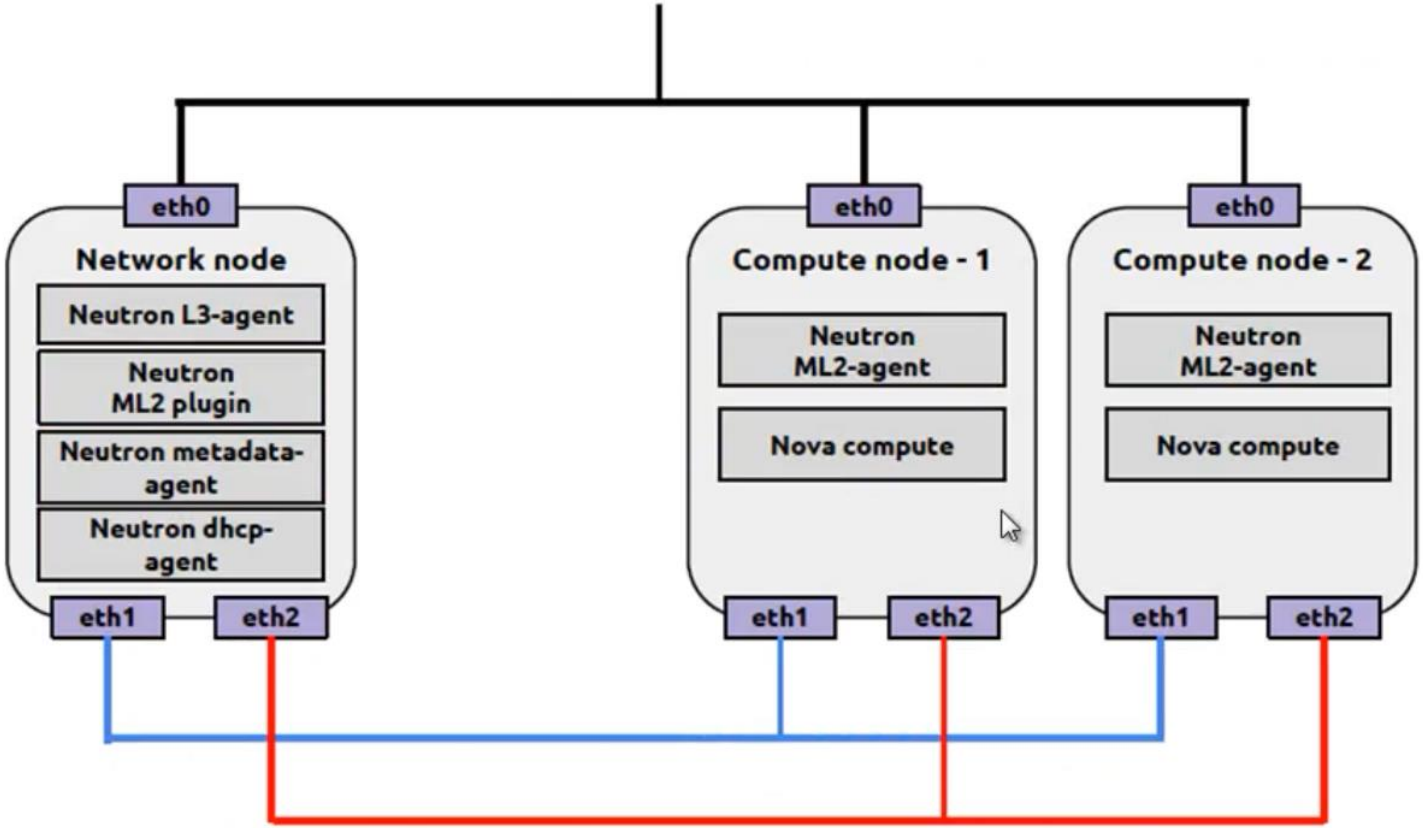
- 负责公网浮动 IP 地址和 NAT
- 负责其他三层特性，例如：负载均衡等
- 每个 Network 对应一个 L3 agent

Metadata-agent

- 提供元数据服务


API Client
Neutron Server
Nova Server
Create Network (POST /tenant1/network)
Network UUID: 'abc'
Create Server (POST /tenant1/server)
Server UUID: 'def'
Get Server Interface(s) (GET /tenant1/server/def/interface)
Server Interface UUID List: ['ghi']
Create Port on Network (POST /tenant1/network/abc/port)
Port UUID 'jkl'
Attach Interface to port (PUT /tenant1/network/abc/port/jkl) { 'attachment' : 'ghi' }
Success


Neutron
ML2 Plugin
TypeDriver
MechanismDriver
VLAN
GRE
VxLAN
Flat
OpenvSwitch
Hyper-V
OpenDaylight
Arista
Cisco Nexus
pSwitch


eth0
Network node
Neutron L3-agent
Neutron ML2 plugin
Neutron metadata-agent
Neutron dhcp-agent
eth1
eth2
eth0
Compute node - 1
Neutron ML2-agent
Nova compute
eth1
eth2
eth0
Compute node - 2
Neutron ML2-agent
Nova compute
eth1
eth2

Single FLAT Network

Multi FLAT Network

Mixed FLAT and Private Network

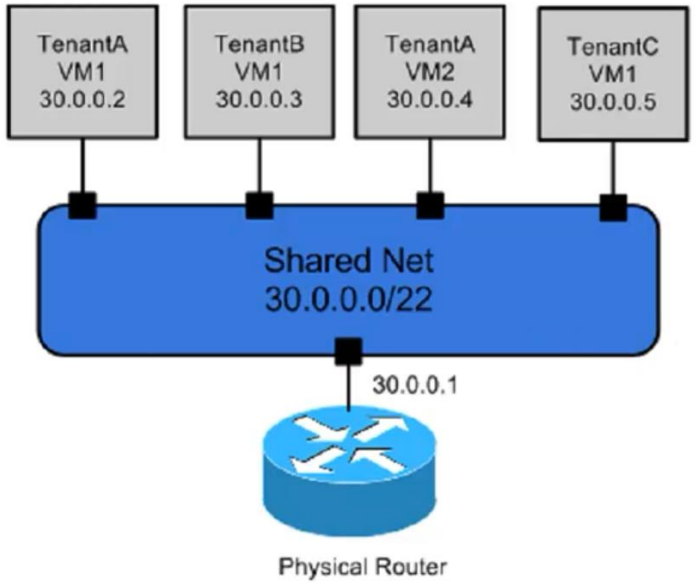
Provider Router with Private Network

Per-tenant Routers with Private Network

类似 Flat Manager

FlatDHCPManager

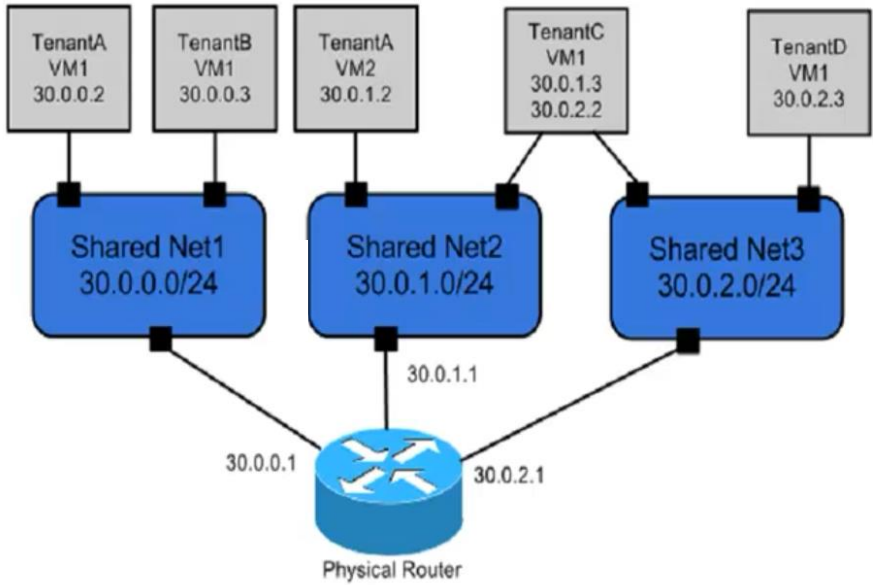
不支持 Floating IP

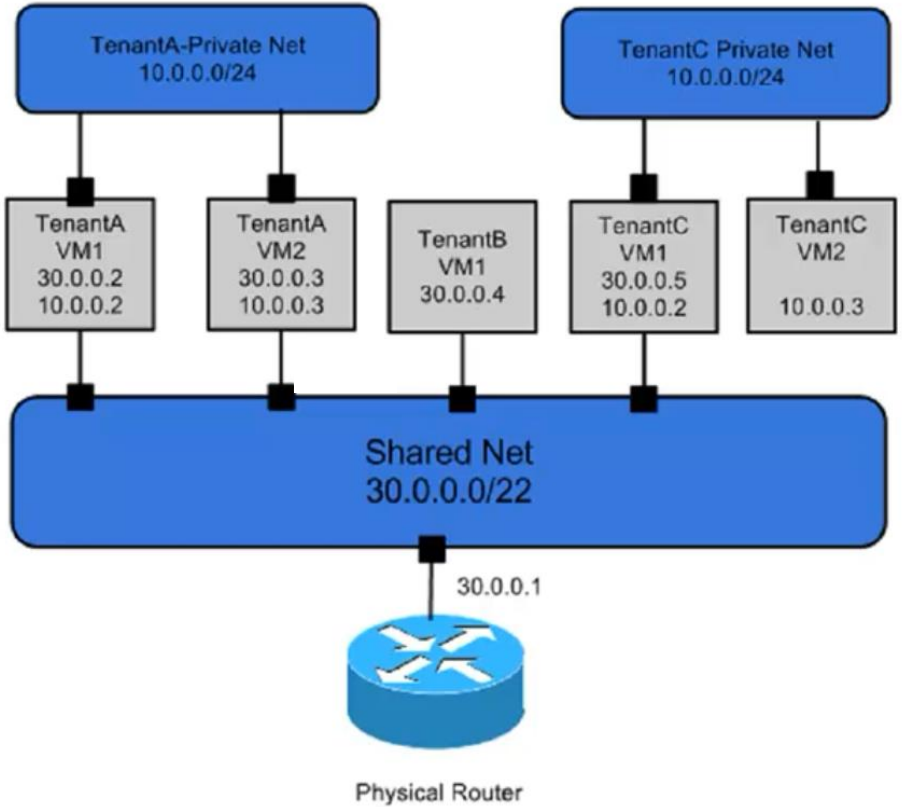


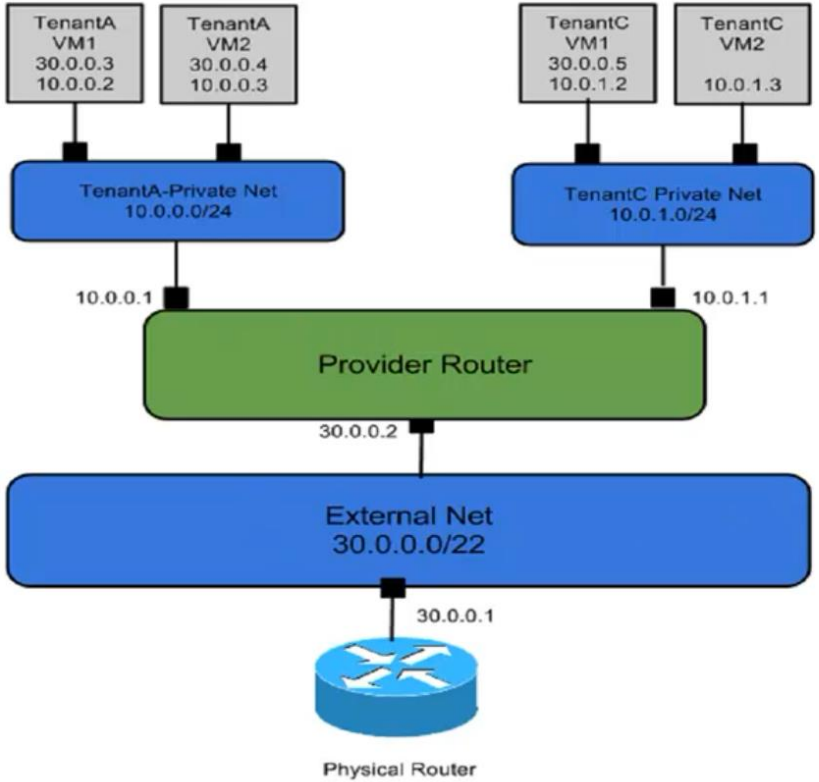
类似 Flat Manager

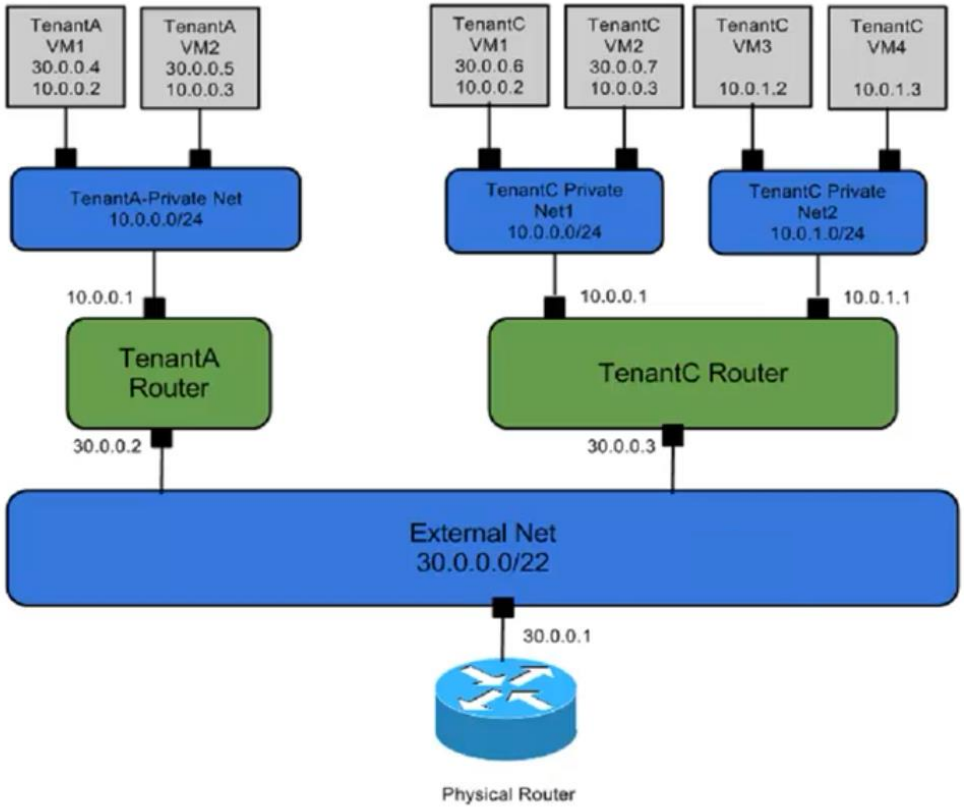
FlatDHCPManager

不支持 Floating IP




TenantA-Private Net
10.0.0.0/24
TenantC Private Net
10.0.0.0/24
TenantA
VM1
30.0.0.2
10.0.0.2
TenantA
VM2
30.0.0.3
10.0.0.3
TenantB
VM1
30.0.0.4
TenantC
VM1
30.0.0.5
10.0.0.2
TenantC
VM2
10.0.0.3
Shared Net
30.0.0.0/22
30.0.0.1
Physical Router


TenantA
VM1
30.0.0.3
10.0.0.2
TenantA
VM2
30.0.0.4
10.0.0.3
TenantC
VM1
30.0.0.5
10.0.1.2
TenantC
VM2
10.0.1.3
TenantA-Private Net
10.0.0.0/24
TenantC Private Net
10.0.1.0/24
10.0.0.1
10.0.1.1
Provider Router
30.0.0.2
External Net
30.0.0.0/22
30.0.0.1
Physical Router


TenantA VM1 30.0.0.4 10.0.0.2
TenantA VM2 30.0.0.5 10.0.0.3
TenantC VM1 30.0.0.6 10.0.0.2
TenantC VM2 30.0.0.7 10.0.0.3
TenantC VM3 10.0.1.2
TenantC VM4 10.0.1.3
TenantA-Private Net 10.0.0.0/24
TenantC Private Net1 10.0.0.0/24
TenantC Private Net2 10.0.1.0/24
10.0.0.1
10.0.0.1
10.0.1.1
TenantA Router
TenantC Router
30.0.0.2
30.0.0.3
External Net 30.0.0.0/22
30.0.0.1
Physical Router

3 代码构建

代码构建

2.4.7 Horizon – 仪表盘套件

讲师：汪洋

目录

1 先决条件

- 安装OpenStack compute(nova)和identity(keystone) service
- 安装Python2.6或2.7，并必须支持Django
- 你的浏览器必须支持HTML5并启用cookies和JavaScript功能

2 代码构建

代码构建

2.4.8 Cinder – 块存储

讲师：汪洋

目录

1 组件说明

- OpenStack块存储服务为云主机提供块存储设备。支持不同后端
- The Block Storage API和scheduler服务运行在controller节点
- The volume service运行在一个或多个存储节点
- 存储节点可以通过本地磁盘、 SAN/NAS等后端设备为云主机提供卷存储

- cinder-api 允许API请求，并路由他们到cinder-volume
- cinder-volume 直接与块存储服务交互。处理像 cinder-scheduler 这样的服务。通过消息队列相互通信。支持多种存储类型
- cinder-scheduler daemon 选择最优的存储节点创建卷。类似于novascheduler
- Messagin queue 在快存储进程中传递消息。

2 代码构建

代码构建