

基于恒天云 3.8 源码安装 neutron

周威光整理

2017-06-28

目 录

1	环境准备	3
1.1	物理环境	3
1.2	软件环境	4
2	openvswitch 和 neutron 源码安装	4
2.1	openvswitch2.4.0 源码安装	4
2.2	neutron 源码安装	5
2.3	额外软件包安装	8
2.4	兼容 neutron 恒天云代码修改部分	8
2.5	根据 neutron 配置文件填充数据库	9
2.6	配置外网网卡桥接	9
2.7	注册 neutron 各服务管理脚本并启动各服务	11

1 环境准备

1.1 物理环境

- (1). 两台 3 网卡的物理机，一台作为控制节点 hty-controller，一台作为计算节点 hty-compute1
- (2). 提前在两台机器上用恒天云 3.8 分别安装控制节点和计算节点
- (3). 每台物理机的一张网卡 eth0 访问外网，一张网卡 eth1 作为管理网络的网卡，一张网卡 eth2 作为 neutron 隧道网络的网卡
- (4). 下面是控制节点和计算节点的网络配置：
控制节点如下：

```
1  # This file describes the network interfaces available on your system
2  # and how to activate them. For more information, see interfaces(5).
3
4  # The loopback network interface
5  auto lo
6  iface lo inet loopback
7
8  # The primary network interface
9  auto eth0
10  iface eth0 inet static
11  address 172.16.19.147
12  netmask 255.255.255.0
13  gateway 172.16.19.254
14  dns-nameservers 172.16.5.1
15
16  auto eth1
17  iface eth1 inet static
18  address 10.10.10.11
19  netmask 255.255.255.0
20
21  auto eth2
22  iface eth2 inet static
23  address 10.10.11.11
24  netmask 255.255.255.0
```

计算节点如下：

```
1  # This file describes the network interfaces available on your system
2  # and how to activate them. For more information, see interfaces(5).
3
4  # The loopback network interface
5  auto lo
6  iface lo inet loopback
7
8  # The primary network interface
9  auto eth0
```

```
10     iface eth0 inet static
11         address 172.16.19.151
12         netmask 255.255.255.0
13         gateway 172.16.19.254
14         dns-nameservers 172.16.5.1
15
16     auto eth1
17     iface eth1 inet static
18         address 10.10.10.12
19         netmask 255.255.255.0
20
21     auto eth2
22     iface eth2 inet static
23         address 10.10.11.12
24         netmask 255.255.255.0
```

1.2 软件环境

- (1). M 版 openstack 的 neutron 源码包：由于恒天云是基于 J 版的 openstack，安装高版本 neutron 会出现兼容性问题。需要在虚拟环境通过源码安装 neutron
- (2). openvswitch2.4.0 源码包：neutron 的 DVR 特性需要 openvswitch2.1 版本以上的支持，直接 apt-get 安装的是低版本的。需要源码安装高版本的。
- (3). openvswitch 和 neutron 各服务启动脚本：源码安装的服务启动比较麻烦，为了方便部署，准备了管理脚本，需将其拷贝到/etc/init.d，并注册。

2 openvswitch 和 neutron 源码安装

2.1 openvswitch2.4.0 源码安装

1. 下载 openvswitch 源码 <http://openvswitch.org/download/>，选择的是 openvswitch2.4.0
2. 解压安装包:tar -xzf openvswitch-2.4.0.tar.gz
3. 构建基于 Linux 内核的交换机，uname -r 用来得到自己 linux 内核版本号，./configure 部分可以用--prefix= 参数，可以让 OVS 完全安装在该目录下。
命令如下：

```
1  cd openvswitch-2.4.0
2  aptitude install dh-autoreconf libssl-dev openssl          #预先安装一些库
3  ./configure --with-linux=/lib/modules/$(uname -r)/build
```

4. 编译并安装 openvswitch-2.4.0

命令如下:

```
1 make & make install
```

5. 安装并加载构建的内核模块

命令如下:

```
1 modprobe libcrc32c
2 modprobe gre
3 modprobe vxlan
4 modprobe openvswitch
5 make modules_install
6 insmod datapath/linux/openvswitch.ko
```

此时可以通过 `lsmod |grep openvswitch` 来查看已载入系统的模块, 发现有 OpenvSwitch

6. 使用 ovsdb 工具初始化配置数据库

命令如下:

```
1 ovsdb-tool create /usr/local/etc/openvswitch/conf.db /usr/local/share/
  openvswitch/vswitch.ovsschema
```

7. 将准备好的 openvswitch 脚本复制到/etc/init.d/, 使用如下命令注册服务, 并使用注册的服务启动 openvswitch

```
1 update-rc.d openvswitch defaults
2 mkdir /var/log/openvswitch #创建存放日志文件夹, 否则启动会报错
3 service openvswitch start
4 ln -s /usr/local/bin/ovs-* /usr/bin/ #创建软连接
```

8. 手动添加一个虚拟网桥, 用于 neutron 服务使用 `ovs-vsctl add-br br-ex`

9. 向/etc/modules 写入如下参数, 使得开机自动加载内核模块

命令如下:

```
1 echo "openvswitch" >> /etc/modules
2 echo "gre" >> /etc/modules
3 echo "vxlan" >> /etc/modules
4 echo "libcrc32c" >> /etc/modules
```

2.2 neutron 源码安装

1. 需要安装 pip 版本在 8.1.2 之上, 否则可能会报错

```
1 apt-get install python-pip
2 pip install --upgrade 'pip>=8.1.2'
```

2. 安装 virtualenv, 用于创建虚拟环境

```
1 pip install virtualenv
```

3. 创建名为 venv 的虚拟环境用于安装 neutron, 并切换进 venv 目录

```
1 virtualenv venv
2 cd venv
```

安装之后使用 ls 命令, 查看当前文件夹的内容, 会多出来 bin,lib,local 等文件夹。以后安装的包会放在这些文件夹, 将不会和系统的混合在一起。这就是隔离的意义。

4. 启动虚拟环境

```
1 source bin/activate
```

5. 切换到 neutron 源码包, 安装 neutron 依赖包。要先安装 python2.7-dev, 再安装 neutron 依赖包, 否则可能会报错

```
1 sudo apt-get install python2.7-dev
2 pip install -r requirements.txt
```

6. 安装 neutron 组件, 要先安装 git, 再安装 neutron 组件, 否则可能会报错

```
1 apt-get install git
2 python setup.py install
```

7. 通过源码的 tools 文件夹下的 generate_config_file_samples.sh 脚本生成配置文件, 并拷贝到/etc/neutron 文件夹下 ps: 配置文件可以从已有的环境直接拷贝过来, 可以节省许多时间

```
1 ./tools/generate_config_file_samples.sh
```

8. 在 mysql 创建 neutron 数据库, 并使用 keystone 命令生成 neutron 的 service 和 endpoint 端点

ps: 只需要在控制节点执行如下操作

```
1 #创建 neutron 数据库
2 mysql -uroot -phtYun@2014
3 CREATE DATABASE neutron;
```

```

4 GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' \
5 IDENTIFIED BY 'NEUTRON_DBPASS';
6 GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' \
7 IDENTIFIED BY 'NEUTRON_DBPASS';
8 #创建 neutron 用户
9 keystone user-create --name neutron --pass neutron_pass@2014
10 keystone user-role-add --user neutron --tenant service --role admin
11 #创建 neutron 服务
12 keystone service-create --name neutron --type network \
13 --description "OpenStack Networking"
14 #创建 neutron 服务端点 endpoint
15 keystone endpoint-create \
16 --service-id $(keystone service-list | awk '/ network / {print $2}') \
17 --publicurl http://hty-controller:9696 \
18 --adminurl http://hty-controller:9696 \
19 --internalurl http://hty-controller:9696 \
20 --region regionOne

```

9. 修改/etc/nova/nova.conf，主要是注释掉 nova-network 的配置，加入 neutron 配置，其余部分不用修改。控制节点参考 172.16.19.147，计算节点参考 172.16.19.151。nova.conf 文件修改如下：

```

1 [DEFAULT]
2 ...
3
4 #下面注释nova-network的配置
5 #network_api_class = nova.network.api.API
6 #security_group_api = nova
7 #firewall_driver = nova.virt.libvirt.firewall.IptablesFirewallDriver
8 #firewall_driver = nova.virt.firewall.NoopFirewallDriver
9
10 #network_manager = nova.network.manager.FlatDHCPManager
11 #network_manager = nova.network.manager.VlanManager
12
13 #启用neutron配置
14 network_api_class = nova.network.neutronv2.api.API
15 security_group_api = neutron
16 linuxnet_interface_driver = nova.network.linux_net.LinuxOVSIfaceDriver
17 firewall_driver = nova.virt.firewall.NoopFirewallDriver
18
19 use_neutron = True
20 ...
21
22 [glance] #保持原有配置
23 ...
24
25 [keystone_authtoken] #保持原有配置
26 ...
27
28 [neutron] #新增
29 url = http://hty-controller:9696
30 auth_strategy = keystone
31 admin_auth_url = http://hty-controller:35357/v2.0
32 admin_tenant_name = service

```

```

33 admin_username = neutron
34 admin_password = neutron_pass@2014
35
36 [database] #保持原有配置
37 ...
38
39 [libvirt] #保持原有配置
40 ...

```

上面配置可能会有遗漏，建议参考 172.16.19.147

10. 复制/etc/neutron 文件夹下的各服务配置文件，控制节点复制 172.16.19.147，计算节点复制 172.16.19.151。这里不再赘述。

ps: 也可以通过源码脚本生成各配置文件，参考官方 neutron 配置，手动配置。会比较麻烦。

11. 如果在步骤 10 直接拷贝的，此处可以忽略。如果手动参照官方配置，需要注意控制节点/etc/neutron/rootwrap.d/l3.filters 和 dhcp.filters 这两个文件这是由于使用了虚拟环境的缘故

```

1 #将文件中的下面一行
2 kill_metadata: KillFilter, root, python, -9
3 #改为
4 kill_metadata: KillFilter, root, /root/venv/bin/python, -9

```

2.3 额外软件包安装

neutron 源码安装并不能将服务依赖的其他软件包全部下载，还需要手动下载。下载如下包：

```

1 pip install python-memcached
2 pip install pymysql
3 apt-get install arping
4 ln -s /root/venv/bin/neutron-ns-metadata-proxy /usr/bin/
5 apt-get install conntrack
6 apt-get install dnsmasq-utils
7 apt-get install ipset

```

2.4 兼容 neutron 恒天云代码修改部分

1. 修改 neutron/common/config.py 文件

```

1 vi /root/venv/local/lib/python2.7/site-packages/neutron/common/config.py
2 #修改 config.py 文件中的如下函数
3 def set_db_defaults():
4     # Update the default QueuePool parameters. These can be tweaked by the

```



```

5      # conf variables — max_pool_size, max_overflow and pool_timeout
6      db_options.set_defaults(
7          cfg.CONF,
8          connection='sqlite://',
9          #sqlite_db='', max_pool_size=10, #注释掉这一行
10         max_pool_size=10,          #添加这一行
11         max_overflow=20, pool_timeout=10)

```

2. 修改 servers.py 文件

```

1      vi /usr/lib/python2.7/dist-packages/nova/api/openstack/compute/servers.py
2      #搜索关键字floating_ips修改floating_ips的语句如下:
3      floating_ips = body['server'].get('fips', None)

```

3. 修改 neutron/agent/common/config.py 文件

```

1      vi /root/venv/lib/python2.7/site-packages/neutron/agent/common/config.py
2      #搜索关键字verbose修改verbose的条件语句如下:
3      def get_log_args(conf, log_file_name, **kwargs):
4          ...
5      #if conf.verbose:          #注释这一行
6      if getattr(conf, "verbose", False): #新增这一行
7          ...

```

2.5 根据 neutron 配置文件填充数据库

```

1  neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/
    neutron/plugins/ml2/ml2_conf.ini upgrade head

```

2.6 配置外网网卡桥接

租户网络内的 vm 若要访问外网, 需要将外网网卡桥接到 br-ex 有如下两部分修改

1. 命令行进行如下操作:

```

1      ip link set eth0 address br-ex 的 mac
2      ovs-vsctl add-port br-ex eth0

```

2. 修改网卡配置如下控制节点如下:

```

1      # This file describes the network interfaces available on your system
2      # and how to activate them. For more information, see interfaces(5).
3
4      # The loopback network interface

```

```

5      auto lo
6      iface lo inet loopback
7
8      # The primary network interface
9      #auto eth0
10     #iface eth0 inet static
11     #address 172.16.19.147
12     #netmask 255.255.255.0
13     #gateway 172.16.19.254
14     #dns-nameservers 172.16.5.1
15
16     auto br-ex
17     iface br-ex inet static
18     address 172.16.19.147
19     netmask 255.255.255.0
20     gateway 172.16.19.254
21     dns-nameservers 8.8.8.8
22
23     auto eth0
24     iface eth0 inet manual
25     up ifconfig $IFACE 0.0.0.0 up
26     up ip link set $IFACE promisc on
27     down ip link set $IFACE promisc off
28     down ifconfig $IFACE down
29
30     auto eth1
31     iface eth1 inet static
32     address 10.10.10.11
33     netmask 255.255.255.0
34
35     auto eth2
36     iface eth2 inet static
37     address 10.10.11.11
38     netmask 255.255.255.0

```

计算节点如下:

```

1      # This file describes the network interfaces available on your system
2      # and how to activate them. For more information, see interfaces(5).
3
4      # The loopback network interface
5      auto lo
6      iface lo inet loopback
7
8      # The primary network interface
9      #auto eth0
10     #iface eth0 inet static
11     #address 172.16.19.151
12     #netmask 255.255.255.0
13     #gateway 172.16.19.254
14     #dns-nameservers 172.16.5.1
15
16     auto br-ex
17     iface br-ex inet static
18     address 172.16.19.151

```

```
19 netmask 255.255.255.0
20 gateway 172.16.19.254
21 dns-nameservers 8.8.8.8
22
23 auto eth0
24 iface eth0 inet manual
25 up ifconfig $IFACE 0.0.0.0 up
26 up ip link set $IFACE promisc on
27 down ip link set $IFACE promisc off
28 down ifconfig $IFACE down
29
30 auto eth1
31 iface eth1 inet static
32 address 10.10.10.12
33 netmask 255.255.255.0
34
35 auto eth2
36 iface eth2 inet static
37 address 10.10.11.12
38 netmask 255.255.255.0
```

更改网络配置，重启网卡，会出现外网无法访问

2.7 注册 neutron 各服务管理脚本并启动各服务

```
1 #复制脚本到/etc/init.d/
2 cp neutron-server neutron-openvswitch-agent neutron-l3-agent neutron-dhcp-agent
   neutron-metadata-agent /etc/init.d/
3 #注册服务
4 update-rc.d neutron-server defaults
5 update-rc.d neutron-openvswitch-agent defaults
6 update-rc.d neutron-l3-agent defaults
7 update-rc.d neutron-dhcp-agent defaults
8 update-rc.d neutron-metadata-agent defaults
9 #启动服务，控制节点全部启动，计算节点只需要启动
10 #neutron-openvswitch-agent, neutron-l3-agent, neutron-metadata-agent
11 mkdir /var/log/neutron #创建neutron日志文件夹，否则启动会报错
12 service neutron-server start
13 service neutron-openvswitch-agent start
14 service neutron-l3-agent start
15 service neutron-dhcp-agent start
16 service neutron-metadata-agent start
```