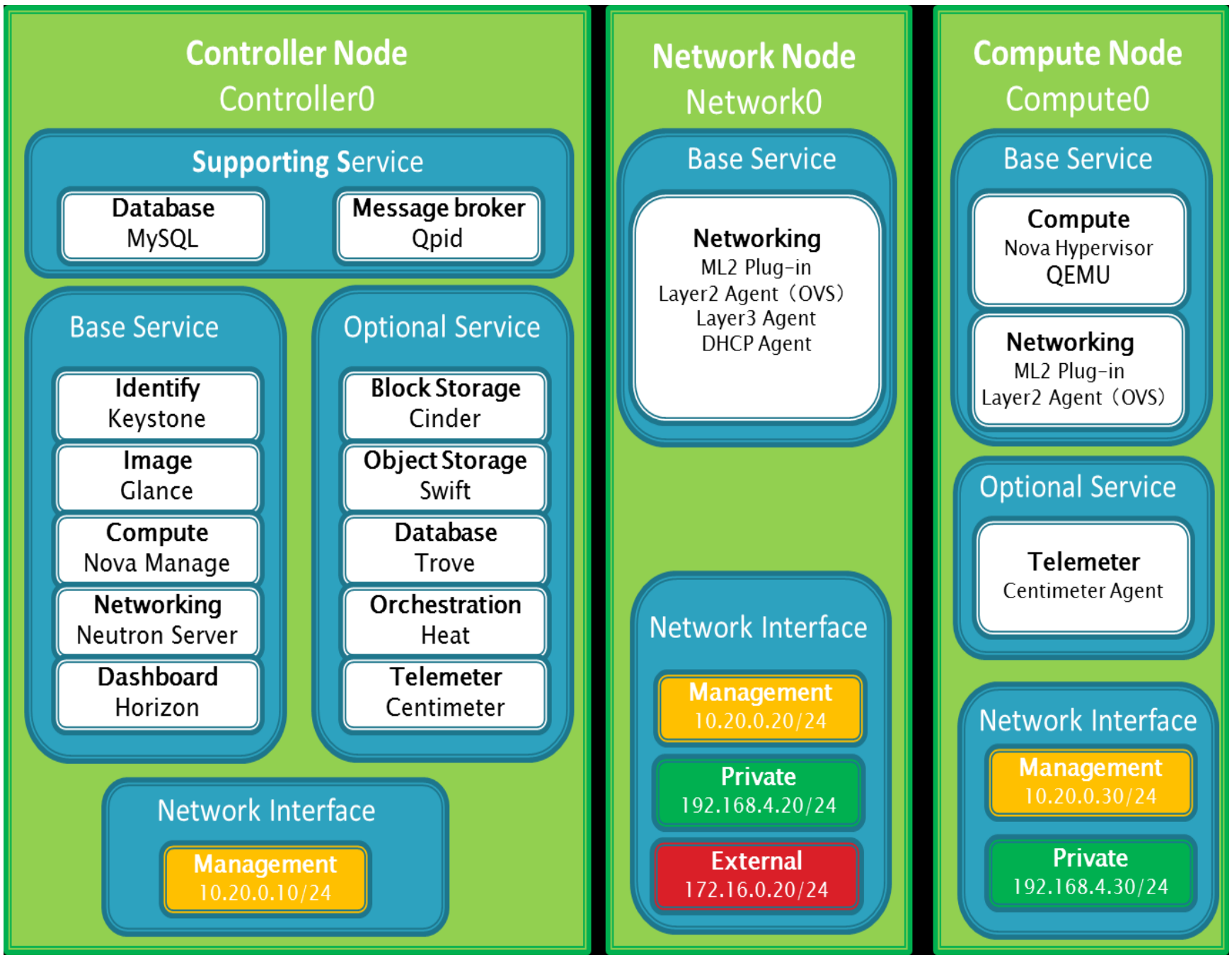
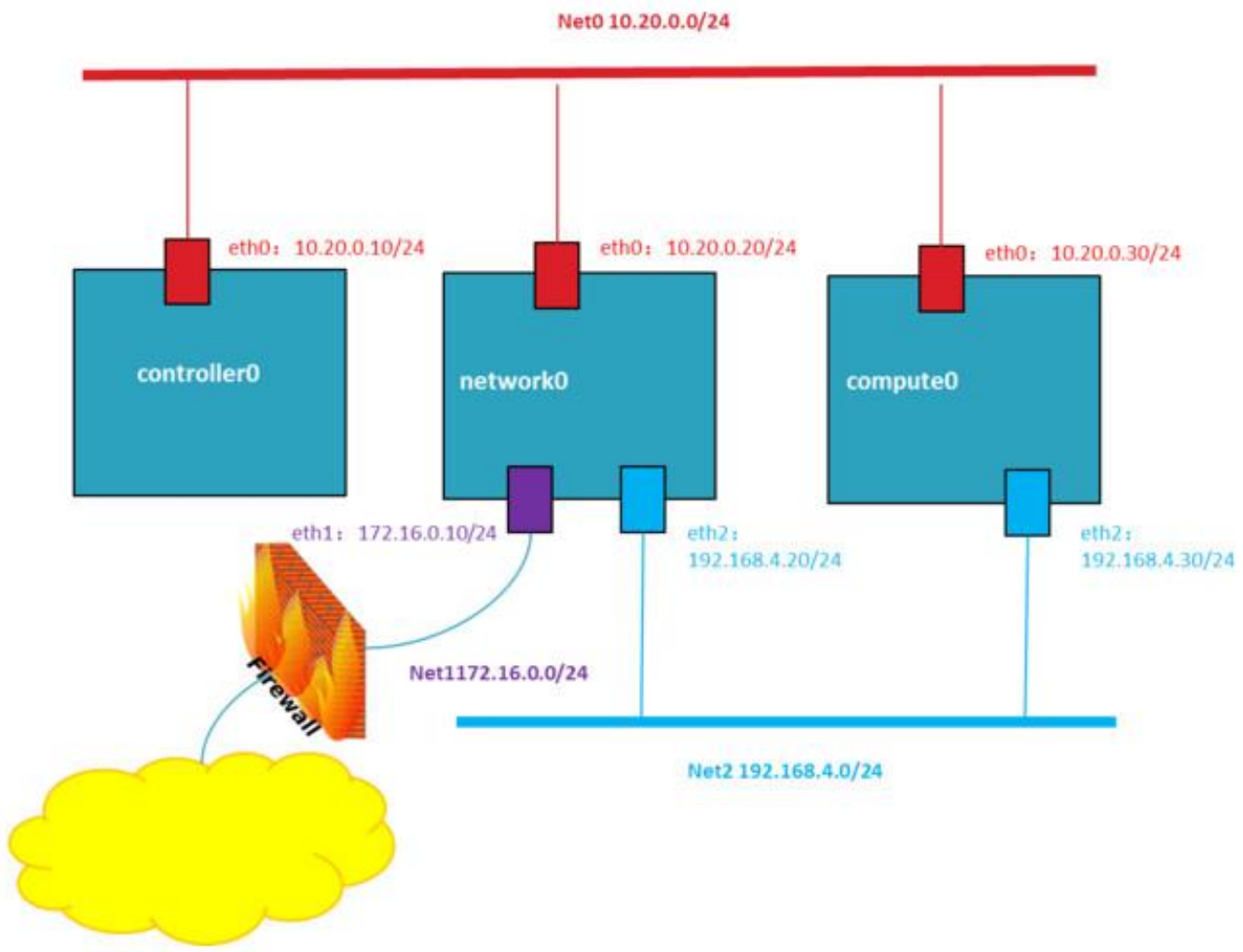
一.部署架构

为了更好的展现 OpenStack 各组件分布式部署的特点，以及逻辑网络配置的区别，本实验不采用 All in One 的部署模式，而是采用多节点分开部署的方式，方便后续学习研究。



二.网络拓扑



三.环境准备

本实验采用 VMware Workstation Windows 版作为虚拟化平台，模拟相应的物理网络和物理服务器，如果需要部署到真实的物理环境，此步骤可以直接替换为在物理机上相应的配置，其原理相同。

3.1 虚拟网络

需要新建 3 个虚拟网络 Net0、Net1 和 Net2，其在 virtual box 中对应配置如下。

```
Net2:
    Network name: VirtualBox  host-only Ethernet Adapter#2
    Purpose: administrator / management network
    IP block: 10.20.0.0/24
    DHCP: disable
    Linux device: eth0

Net3:
    Network name: VirtualBox  host-only Ethernet Adapter#3
    Purpose: public network
    DHCP: disable
    IP block: 172.16.0.0/24
    Linux device: eth1

Net4:
    Network name: VirtualBox  host-only Ethernet Adapter#4
    Purpose: Storage/private network
    DHCP: disable
    IP block: 192.168.4.0/24
    Linux device: eth2
```

3.2 虚拟机

需要新建 3 个虚拟机 VM0、VM1 和 VM2，其对应配置如下。

```
controller0
     eth0:10.20.0.10   (management network)
     eht1:(disabled)
     eht2:(disabled)

network0
     eth0:10.20.0.20   (management network)
     eht1:172.16.0.20   (public/external network)
     eht2:192.168.4.20  (private network)

compute0
     eth0:10.20.0.30   (management network)
     eht1:(disabled)
     eht2:192.168.4.30  (private network)

compute1  (optional)
     eth0:10.20.0.31   (management network)
     eht1:(disabled)
     eht2:192.168.4.31  (private network)
```

3.3 网络设置

```
controller0
     eth0:10.20.0.10   (management network)
     eht1:(disabled)
     eht2:(disabled)

network0
     eth0:10.20.0.20   (management network)
     eht1:172.16.0.20   (public/external network)
     eht2:192.168.4.20  (private network)

compute0
     eth0:10.20.0.30   (management network)
     eht1:(disabled)
     eht2:192.168.4.30  (private network)

compute1  (optional)
```

```
    eth0:10.20.0.31   (management network)
    eht1:(disabled)
    eht2:192.168.4.31  (private network)
```

3.4 操作系统准备

本实验使用 Linux 发行版 CentOS 6.5 x86_64，在安装操作系统过程中，选择的初始安装包为“基本”安装包。

使用 virtualbox 中网络设置如下

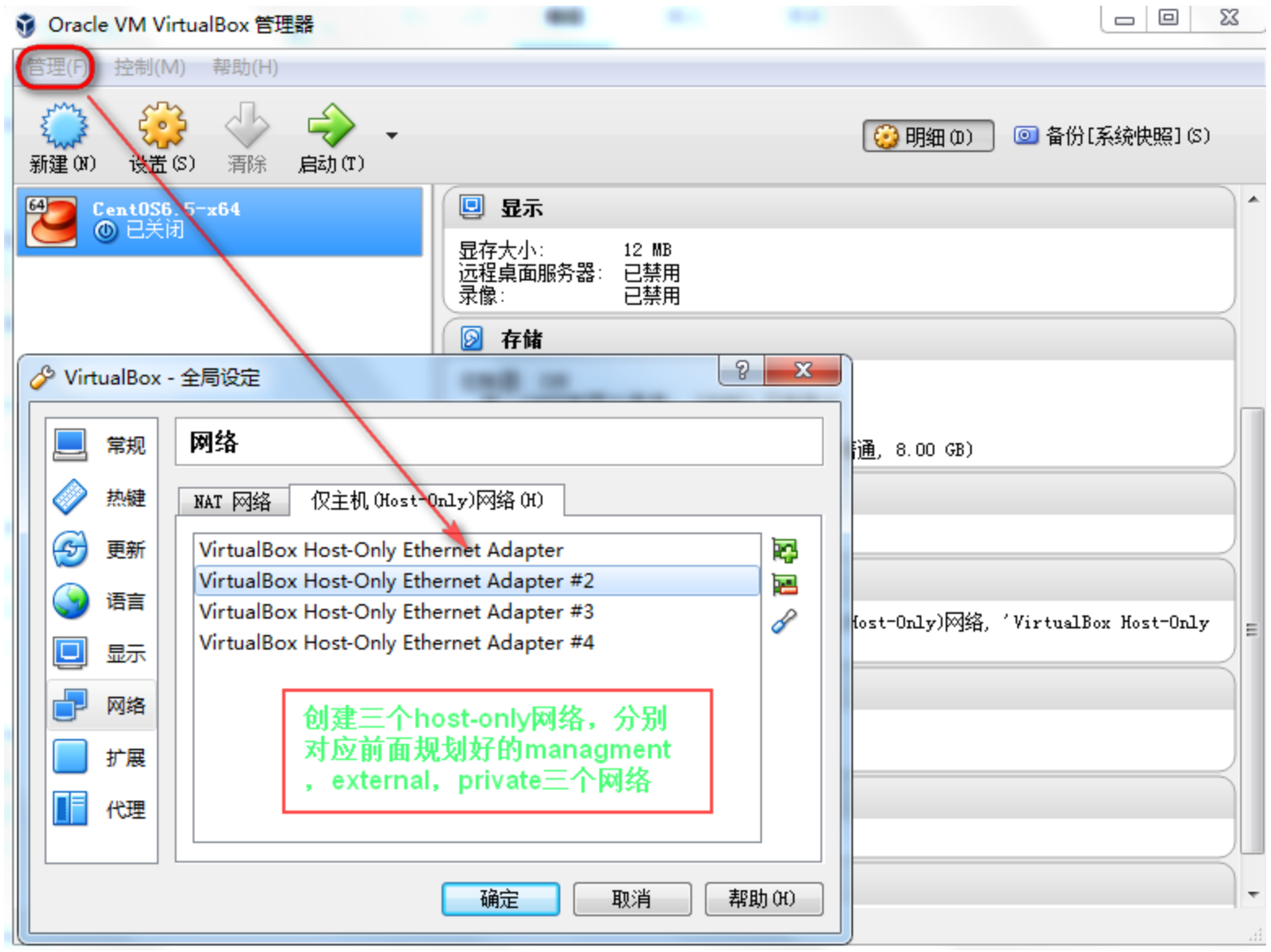


图 5 三个网段的创建

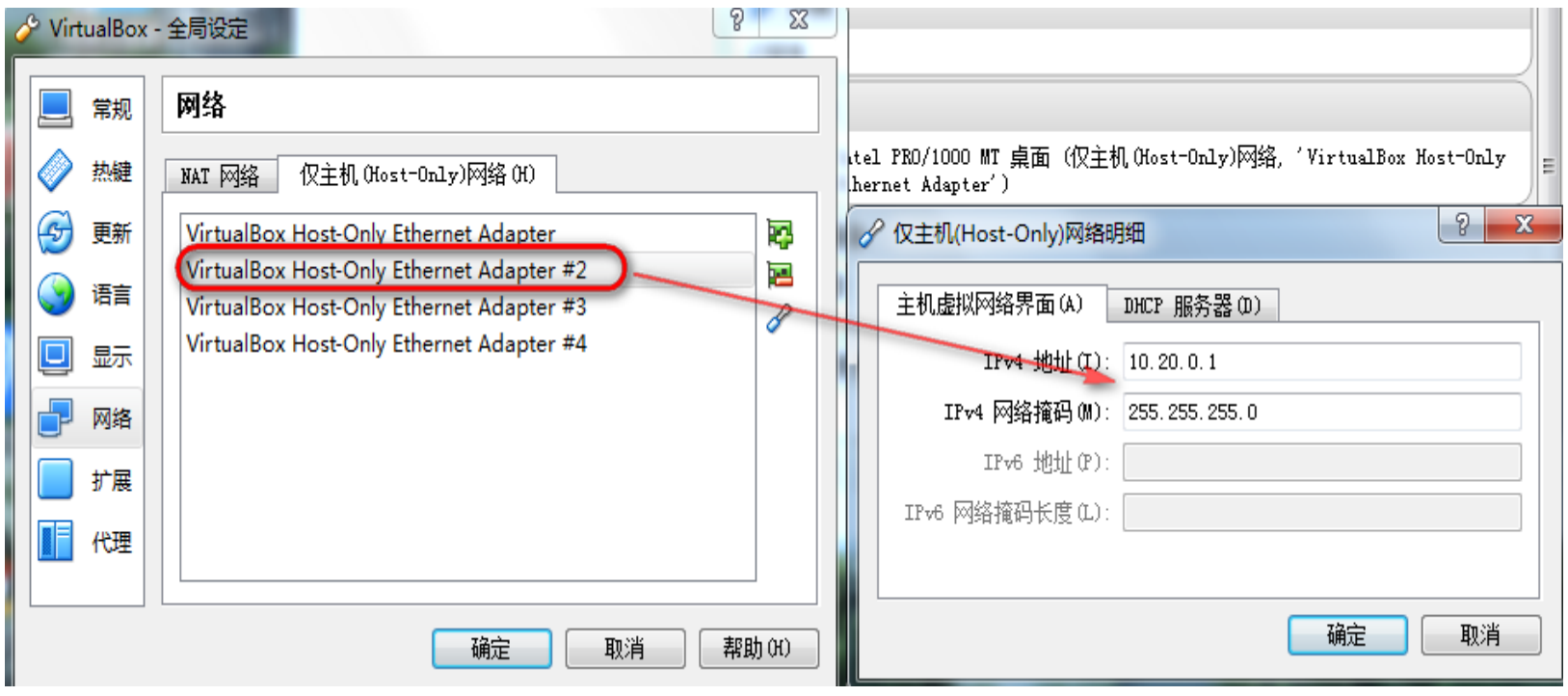


图 6 三个网络的设置

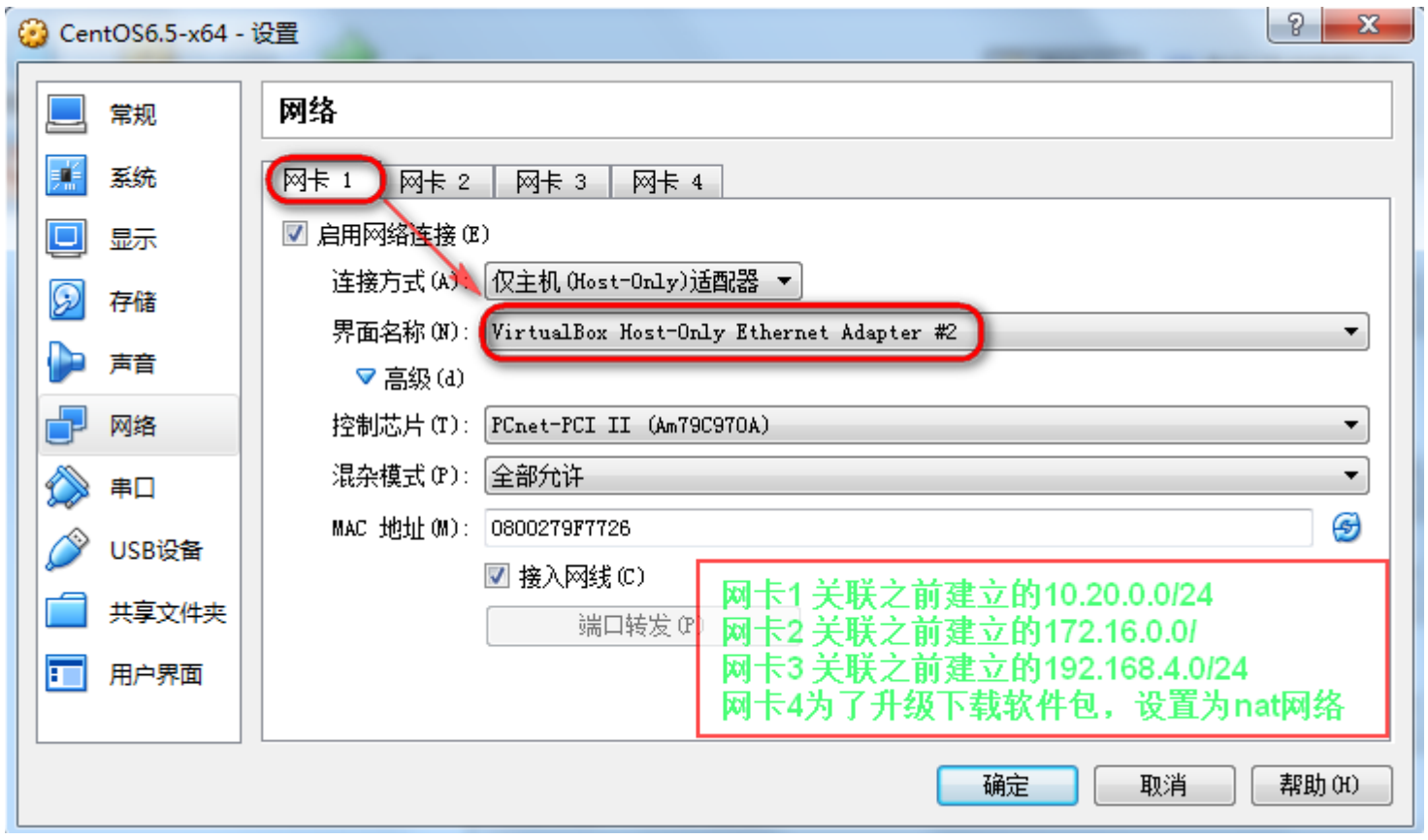


图 7 几台虚拟机的网络设置

安装完成系统以后还需要额外配置如下 YUM 仓库。

ISO 文件下载：http://mirrors.163.com/centos/6.5/isos/x86_64/CentOS-6.5-x86_64-bin-DVD1.iso

EPEL 源: http://dl.fedoraproject.org/pub/epel/6/x86_64/

RDO 源: https://repos.fedorapeople.org/repos/openstack/EOL/openstack-icehouse/epel-6/

自动配置执行如此命令即可，源安装完成后更新所有 RPM 包，由于升级了 kernel 需要重新启动操作系统。

```
yum install -y https://repos.fedorapeople.org/repos/openstack/EOL/openstack-icehouse/rdo-release-icehou
se-4.noarch.rpm
yum install -y http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
yum update -y
reboot -h 0
```

接下来可以开始安装配置

四.公共配置（all nodes）

4.1 以下命令需要在每一个节点都执行

修改 hosts 文件

```
vi /etc/hosts
127.0.0.1    localhost
::1          localhost
10.20.0.10   controller0
10.20.0.20   network0
10.20.0.30   compute0
```

禁用 selinux

```
vi /etc/selinux/config
SELINUX=disabled
```

安装 NTP 服务

```
yum install ntp -y
service ntpd start
chkconfig ntpd on
```

修改 NTP 配置文件，配置从 controller0 时间同步。(除了 controller0 以外)

```
vi /etc/ntp.conf

server 10.20.0.10
fudge  10.20.0.10 stratum 10  # LCL is unsynchronized
```

立即同步并检查时间同步配置是否正确。(除了 controller0 以外)

```
ntpdate -u 10.20.0.10
service ntpd restart
ntpq -p
```

清空防火墙规则

```
Iptables –F
Iptables –t nat -F
Service iptables save
```

重启防火墙，查看是否生效

```
Service iptables restart
Iptables -L
```

安装 openstack-utils,方便后续直接可以通过命令行方式修改配置文件（下面使用的 openstack-set 命令工具，使用它不用修改各组件的配置文件就可以使组件配置变更）

```
yum install -y openstack-utils
```

4.2 基本服务安装与配置（controller0 node）

基本服务包括 NTP 服务、MySQL 数据库服务和 AMQP 服务，本实例采用 MySQL 和 Qpid 作为这两个服务的实现。

修改 NTP 配置文件，配置从 127.127.1.0 时间同步。

```
vi /etc/ntp.conf
server 127.127.1.0
```

重启 ntp service

```
service ntpd restart
```

MySQL 服务安装

```
yum install -y mysql mysql-server MySQL-python
```

修改 MySQL 配置

```
vi /etc/my.cnf
[mysqld]
bind-address = 0.0.0.0
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
```

启动 MySQL 服务

```
service mysqld start
chkconfig mysqld on
```

交互式配置 MySQL root 密码，设置密码为“openstack”

```
mysql_secure_installation
```

Qpid 安装消息服务，设置客户端不需要验证使用服务

```
1    yum install -y qpid-cpp-server
2
3    vi /etc/qpidd.conf
4    auth=no
```

配置修改后，重启 Qpid 后台服务

```
yum install -y qpid-cpp-server

vi /etc/qpidd.conf
auth=no
```

4.3 控制节点安装（controller0）

主机名设置

```
vi /etc/sysconfig/network
HOSTNAME=controller0
```

网卡配置

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=10.20.0.10
NETMASK=255.255.255.0
```

网络配置文件修改完后重启网络服务

```
serice network restart
```

五.Keyston 安装与配置

安装 keystone 包

```
yum install openstack-keystone python-keystoneclient -y
```

为 keystone 设置 admin 账户的 token

```
ADMIN_TOKEN=$(openssl rand -hex 10)
echo $ADMIN_TOKEN
openstack-config --set /etc/keystone/keystone.conf DEFAULT admin_token $ADMIN_TOKEN
```

配置数据连接

```
openstack-config --set /etc/keystone/keystone.conf sql connection mysql://keystone:openstack@controller
0/keystone
openstack-config --set /etc/keystone/keystone.conf DEFAULT debug True
openstack-config --set /etc/keystone/keystone.conf DEFAULT verbose True
```

设置 Keystone 用 PKI tokens

```
keystone-manage pki_setup --keystone-user keystone --keystone-group keystone
chown -R keystone:keystone /etc/keystone/ssl
chmod -R o-rwx /etc/keystone/ssl
```

为 Keystone 建表

```
mysql -uroot -popenstack -e "CREATE DATABASE keystone;"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY
 'openstack';"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'controller0' IDENTIFIED
BY 'openstack';"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'openst
ack';"
```

初始化 Keystone 数据库

```
su -s /bin/sh -c "keystone-manage db_sync"
```

也可以直接用 openstack-db 工具初始数据库

```
openstack-db --init --service keystone --password openstack
```

启动 keystone 服务

```
service openstack-keystone start
chkconfig openstack-keystone on
```

设置认证信息

```
export OS_SERVICE_TOKEN=`echo $ADMIN_TOKEN`
export OS_SERVICE_ENDPOINT=http://controller0:35357/v2.0
```

创建管理员和系统服务使用的租户

```
keystone tenant-create --name=admin --description="Admin Tenant"
keystone tenant-create --name=service --description="Service Tenant"
```

创建管理员用户

```
keystone user-create --name=admin --pass=admin --email=admin@example.com
```

创建管理员角色

```
keystone role-create --name=admin
```

为管理员用户分配"管理员"角色

```
keystone user-role-add --user=admin --tenant=admin --role=admin
```

为 keystone 服务建立 endpoints

```
keystone service-create --name=keystone --type=identity --description="Keystone Identity Service"
```

为 keystone 建立 servie 和 endpoint 关联

```
keystone endpoint-create \
--service-id=$(keystone service-list | awk '/ identity / {print $2}') \
--publicurl=http://controller0:5000/v2.0 \
--internalurl=http://controller0:5000/v2.0 \
--adminurl=http://controller0:35357/v2.0
```

验证 keystone 安装的正确性取消先前的 Token 变量，不然会干扰新建用户的验证。

```
unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
```

然后用设置环境变量认证,保存认证信息

```
vi ~/keystonerc

export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_TENANT_NAME=admin
export OS_AUTH_URL=http://controller0:35357/v2.0
```

source 该文件使其生效

```
source keystonerc
keystone token-get
```

Keystone 安装结束。

六.Glance 安装与配置

安装 Glance 的包

```
yum install openstack-glance python-glanceclient -y
```

配置 Glance 连接数据库

```
openstack-config --set /etc/glance/glance-api.conf DEFAULT sql_connection mysql://glance:openstack@cont
roller0/glance
openstack-config --set /etc/glance/glance-registry.conf DEFAULT sql_connection mysql://glance:openstack
@controller0/glance
```

初始化 Glance 数据库

```
openstack-db --init --service glance --password openstack
```

创建 glance 用户

```
keystone user-create --name=glance --pass=glance --email=glance@example.com
```

并分配 service 角色

```
keystone user-role-add --user=glance --tenant=service --role=admin
```

创建 glance 服务

```
keystone service-create --name=glance --type=image --description="Glance Image Service"
```

创建 keystone 的 endpoint

```
keystone endpoint-create \
--service-id=$(keystone service-list | awk '/ image / {print $2}')  \
--publicurl=http://controller0:9292 \
--internalurl=http://controller0:9292 \
--adminurl=http://controller0:9292
```

用 openstack-util 修改 glance-api 和 register 配置文件

```
openstack-config --set /etc/glance/glance-api.conf DEFAULT debug True
openstack-config --set /etc/glance/glance-api.conf DEFAULT verbose True
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken auth_uri http://controller0:5000
```

```
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_user glance
openstack-config --set /etc/glance/glance-api.conf keystone_authtoken admin_password glance
openstack-config --set /etc/glance/glance-api.conf paste_deploy flavor keystone

openstack-config --set /etc/glance/glance-registry.conf DEFAULT debug True
openstack-config --set /etc/glance/glance-registry.conf DEFAULT verbose True
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken auth_uri http://controller0:
5000
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken admin_user glance
openstack-config --set /etc/glance/glance-registry.conf keystone_authtoken admin_password glance
openstack-config --set /etc/glance/glance-registry.conf paste_deploy flavor keystone
```

启动 glance 相关的两个服务

```
service openstack-glance-api start
service openstack-glance-registry start

chkconfig openstack-glance-api on
chkconfig openstack-glance-registry on
```

下载最 Cirros 镜像验证 glance 安装是否成功

```
wget http://cdn.download.cirros-cloud.net/0.3.1/cirros-0.3.1-x86_64-disk.img
glance image-create --progress --name="CirrOS 0.3.1" --disk-format=qcow2  --container-format=ovf --is-p
ublic=true < cirros-0.3.1-x86_64-disk.img
```

查看刚刚上传的 image

```
glance  image-list
```

如果显示相应的 image 信息说明安装成功。

七.Nova 安装与配置

nova 安装与配置

```
yum install -y openstack-nova-api openstack-nova-cert openstack-nova-conductor \
openstack-nova-console openstack-nova-novncproxy openstack-nova-scheduler python-novaclient
```

在 keystone 中创建 nova 相应的用户和服务

```
keystone user-create --name=nova --pass=nova --email=nova@example.com
keystone user-role-add --user=nova --tenant=service --role=admin
```

keystone 注册服务

```
keystone service-create --name=nova --type=compute --description="Nova Compute Service"
```

keystone 注册 endpoint

```
keystone endpoint-create \
--service-id=$(keystone service-list | awk '/ compute / {print $2}')  \
--publicurl=http://controller0:8774/v2/%\(tenant_id\)s \
--internalurl=http://controller0:8774/v2/%\(tenant_id\)s \
--adminurl=http://controller0:8774/v2/%\(tenant_id\)s
```

配置 nova MySQL 连接

```
openstack-config --set /etc/nova/nova.conf database connection mysql://nova:openstack@controller0/nova
```

初始化数据库

```
openstack-db --init --service nova --password openstack
```

配置 nova.conf

```
openstack-config --set /etc/nova/nova.conf DEFAULT debug True
openstack-config --set /etc/nova/nova.conf DEFAULT verbose True
openstack-config --set /etc/nova/nova.conf DEFAULT rpc_backend qpid
openstack-config --set /etc/nova/nova.conf DEFAULT qpid_hostname controller0
```

```
openstack-config --set /etc/nova/nova.conf DEFAULT my_ip 10.20.0.10
openstack-config --set /etc/nova/nova.conf DEFAULT vncserver_listen 10.20.0.10
openstack-config --set /etc/nova/nova.conf DEFAULT vncserver_proxyclient_address 10.20.0.10

openstack-config --set /etc/nova/nova.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_uri http://controller0:5000
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_user nova
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_password nova
```

添加 api-paste.ini 的 Keystone 认证信息

```
openstack-config --set /etc/nova/api-paste.ini filter:authtoken paste.filter_factory keystoneclient.mid
dleware.auth_token:filter_factory
openstack-config --set /etc/nova/api-paste.ini filter:authtoken auth_host controller0
openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin_tenant_name service
openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin_user nova
openstack-config --set /etc/nova/api-paste.ini filter:authtoken admin_password nova
```

启动服务

```
service openstack-nova-api start
service openstack-nova-cert start
service openstack-nova-consoleauth start
service openstack-nova-scheduler start
service openstack-nova-conductor start
service openstack-nova-novncproxy start
```

添加到系统服务

```
chkconfig openstack-nova-api on
chkconfig openstack-nova-cert on
chkconfig openstack-nova-consoleauth on
chkconfig openstack-nova-scheduler on
chkconfig openstack-nova-conductor on
chkconfig openstack-nova-novncproxy on
```

检查服务是否正常

```
nova-manage service list
```

检查进程

```
ps -ef|grep nova
```

八.Neutron server 安装与配置

8.1 安装 Neutron server 相关包

yum install -y openstack-neutron openstack-neutron-ml2 python-neutronclient

在 keystone 中创建 Neutron 相应的用户和服务

```
keystone user-create --name neutron --pass neutron --email neutron@example.com
keystone user-role-add --user neutron --tenant service --role admin
keystone service-create --name neutron --type network --description "OpenStack Networking"
keystone endpoint-create \
--service-id $(keystone service-list | awk '/ network / {print $2}') \
--publicurl http://controller0:9696 \
--adminurl http://controller0:9696 \
--internalurl http://controller0:9696
```

为 Neutron 在 MySQL 建数据库

```
mysql -uroot -popenstack -e "CREATE DATABASE neutron;"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY '
openstack';"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY 'openstac
k';"
mysql -uroot -popenstack -e "GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'controller0' IDENTIFIED BY
 'openstack';
```

配置 MySQL

```
openstack-config --set /etc/neutron/neutron.conf database connection mysql://neutron:openstack@controll
er0/neutron
```

配置 Neutron Keystone 认证

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_uri http://controller0:5000
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_user neutron
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_password neutron
```

配置 Neutron qpid

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT rpc_backend neutron.openstack.common.rpc.impl_
qpid
openstack-config --set /etc/neutron/neutron.conf DEFAULT qpid_hostname controller0
openstack-config --set /etc/neutron/neutron.conf DEFAULT notify_nova_on_port_status_changes True
openstack-config --set /etc/neutron/neutron.conf DEFAULT notify_nova_on_port_data_changes True
openstack-config --set /etc/neutron/neutron.conf DEFAULT nova_url http://controller0:8774/v2
openstack-config --set /etc/neutron/neutron.conf DEFAULT nova_admin_username nova
openstack-config --set /etc/neutron/neutron.conf DEFAULT nova_admin_tenant_id $(keystone tenant-list |
awk '/ service / { print $2 }')
openstack-config --set /etc/neutron/neutron.conf DEFAULT nova_admin_password nova
openstack-config --set /etc/neutron/neutron.conf DEFAULT nova_admin_auth_url http://controller0:35357/v
2.0
```

配置 Neutron ml2 plugin 用 openvswitch

```
ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini
openstack-config --set /etc/neutron/neutron.conf DEFAULT core_plugin ml2
openstack-config --set /etc/neutron/neutron.conf DEFAULT service_plugins router
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 type_drivers gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 tenant_network_types gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 mechanism_drivers openvswitch
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_gre tunnel_id_ranges 1:1000
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup firewall_driver neutron.agen
t.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup enable_security_group True
```

配置 nova 使用 Neutron 作为 network 服务

```
openstack-config --set /etc/nova/nova.conf DEFAULT network_api_class nova.network.neutronv2.api.API
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_url http://controller0:9696
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_auth_strategy keystone
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_tenant_name service
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_username neutron
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_password neutron
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_auth_url http://controller0:35357/v2.0
openstack-config --set /etc/nova/nova.conf DEFAULT linuxnet_interface_driver nova.network.linux_net.Lin
uxOVSInterfaceDriver
openstack-config --set /etc/nova/nova.conf DEFAULT firewall_driver nova.virt.firewall.NoopFirewallDrive
r
openstack-config --set /etc/nova/nova.conf DEFAULT security_group_api neutron
openstack-config --set /etc/nova/nova.conf DEFAULT service_neutron_metadata_proxy true
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_metadata_proxy_shared_secret METADATA_SECRET
```

重启 nova controller 上的服务

```
service openstack-nova-api restart
service openstack-nova-scheduler restart
service openstack-nova-conductor restart
```

启动 Neutron server

```
service neutron-server start
chkconfig neutron-server on
```

8.2 网络节点安装（network0 node）

主机名设置

```
vi /etc/sysconfig/network
HOSTNAME=network0
```

网卡配置

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=10.20.0.20
NETMASK=255.255.255.0

vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=172.16.0.20
NETMASK=255.255.255.0

vi /etc/sysconfig/network-scripts/ifcfg-eth2
DEVICE=eth2
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.4.20
NETMASK=255.255.255.0
```

网络配置文件修改完后重启网络服务

```
serice network restart
```

先安装 Neutron 相关的包

```
yum install -y openstack-neutron openstack-neutron-ml2 openstack-neutron-openvswitch
```

允许 ip forward

```
vi /etc/sysctl.conf
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
```

立即生效

```
sysctl -p
```

配置 Neutron keysone 认证

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_uri http://controller0:5000
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_user neutron
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_password neutron
```

配置 qpid

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT rpc_backend neutron.openstack.common.rpc.impl_
qpid
openstack-config --set /etc/neutron/neutron.conf DEFAULT qpid_hostname controller0
```

配置 Neutron 使用 ml + openvswitch +gre

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT core_plugin ml2
openstack-config --set /etc/neutron/neutron.conf DEFAULT service_plugins router

openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 type_drivers gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 tenant_network_types gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 mechanism_drivers openvswitch
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_gre tunnel_id_ranges 1:1000
```

```
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs local_ip 192.168.4.20
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs tunnel_type gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs enable_tunneling True
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup firewall_driver neutron.agen
t.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup enable_security_group True

ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini
cp /etc/init.d/neutron-openvswitch-agent /etc/init.d/neutronopenvswitch-agent.orig
sed -i 's,plugins/openvswitch/ovs_neutron_plugin.ini,plugin.ini,g' /etc/init.d/neutron-openvswitch-agen
t
```

配置 l3

```
openstack-config --set /etc/neutron/l3_agent.ini DEFAULT interface_driver neutron.agent.linux.interface
.OVSInterfaceDriver
openstack-config --set /etc/neutron/l3_agent.ini DEFAULT use_namespaces True
```

配置 dhcp agent

```
openstack-config --set /etc/neutron/dhcp_agent.ini DEFAULT interface_driver neutron.agent.linux.interfa
ce.OVSInterfaceDriver
openstack-config --set /etc/neutron/dhcp_agent.ini DEFAULT dhcp_driver neutron.agent.linux.dhcp.Dnsmasq
openstack-config --set /etc/neutron/dhcp_agent.ini DEFAULT use_namespaces True
```

配置 metadata agent

```
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT auth_url http://controller0:5000/v2.0
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT auth_region regionOne
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT admin_tenant_name service
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT admin_user neutron
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT admin_password neutron
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT nova_metadata_ip controller0
openstack-config --set /etc/neutron/metadata_agent.ini DEFAULT metadata_proxy_shared_secret METADATA_SE
CRET

service openvswitch start
chkconfig openvswitch on

ovs-vsctl add-br br-int
ovs-vsctl add-br br-ex
ovs-vsctl add-port br-ex eth1
```

修改 eth1 和 br-ext 网络配置

```
vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=none
PROMISC=yes

vi /etc/sysconfig/network-scripts/ifcfg-br-ex

DEVICE=br-ex
TYPE=Bridge
ONBOOT=no
BOOTPROTO=none
```

重启网络服务

```
service network restart
```

为 br-ext 添加 ip

```
ip link set br-ex up
sudo ip addr add 172.16.0.20/24 dev br-ex
```

启动 Neutron 服务

```
service neutron-openvswitch-agent start
service neutron-l3-agent start
service neutron-dhcp-agent start
service neutron-metadata-agent start

chkconfig neutron-openvswitch-agent on
chkconfig neutron-l3-agent on
```

```
chkconfig neutron-dhcp-agent on
chkconfig neutron-metadata-agent on
```

8.3 计算节点安装（compute0 node）

主机名设置

```
vi /etc/sysconfig/network
HOSTNAME=compute0
```

网卡配置

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=10.20.0.30
NETMASK=255.255.255.0

vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=172.16.0.30
NETMASK=255.255.255.0

vi /etc/sysconfig/network-scripts/ifcfg-eth2
DEVICE=eth2
TYPE=Ethernet
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=static
IPADDR=192.168.4.30
NETMASK=255.255.255.0
```

网络配置文件修改完后重启网络服务

```
serice network restart
```

安装 nova 相关包

```
yum install -y openstack-nova-compute
```

配置 nova

```
openstack-config --set /etc/nova/nova.conf database connection mysql://nova:openstack@controller0/nova
openstack-config --set /etc/nova/nova.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_uri http://controller0:5000
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/nova/nova.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_user nova
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/nova/nova.conf keystone_authtoken admin_password nova

openstack-config --set /etc/nova/nova.conf DEFAULT rpc_backend qpid
openstack-config --set /etc/nova/nova.conf DEFAULT qpid_hostname controller0

openstack-config --set /etc/nova/nova.conf DEFAULT my_ip 10.20.0.30
openstack-config --set /etc/nova/nova.conf DEFAULT vnc_enabled True
openstack-config --set /etc/nova/nova.conf DEFAULT vncserver_listen 0.0.0.0
openstack-config --set /etc/nova/nova.conf DEFAULT vncserver_proxyclient_address 10.20.0.30
openstack-config --set /etc/nova/nova.conf DEFAULT novncproxy_base_url http://controller0:6080/vnc_auto
.html
openstack-config --set /etc/nova/nova.conf libvirt virt_type qemu

openstack-config --set /etc/nova/nova.conf DEFAULT glance_host controller0
```

启动 compute 节点服务

```
service libvirtd start
service messagebus start
service openstack-nova-compute start

chkconfig libvirtd on
chkconfig messagebus on
chkconfig openstack-nova-compute on
```

在 controller 节点检查 compute 服务是否启动

```
nova-manage service list
```

多出计算节点服务

```
nova-manage service list
```

安装 neutron ml2 和 openvswitch agent

```
yum install openstack-neutron-ml2 openstack-neutron-openvswitch
```

配置 Neutron Keystone 认证

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT auth_strategy keystone
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_uri http://controller0:5000
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_host controller0
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_protocol http
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken auth_port 35357
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_tenant_name service
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_user neutron
openstack-config --set /etc/neutron/neutron.conf keystone_authtoken admin_password neutron
```

配置 Neutron qpid

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT rpc_backend neutron.openstack.common.rpc.impl_
qpid
openstack-config --set /etc/neutron/neutron.conf DEFAULT qpid_hostname controller0
```

配置 Neutron 使用 ml2 for ovs and gre

```
openstack-config --set /etc/neutron/neutron.conf DEFAULT core_plugin ml2
openstack-config --set /etc/neutron/neutron.conf DEFAULT service_plugins router

openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 type_drivers gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 tenant_network_types gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2 mechanism_drivers openvswitch
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ml2_type_gre tunnel_id_ranges 1:1000
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs local_ip 192.168.4.30
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs tunnel_type gre
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini ovs enable_tunneling True
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup firewall_driver neutron.agen
t.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
openstack-config --set /etc/neutron/plugins/ml2/ml2_conf.ini securitygroup enable_security_group True

ln -s /etc/neutron/plugins/ml2/ml2_conf.ini /etc/neutron/plugin.ini
cp /etc/init.d/neutron-openvswitch-agent /etc/init.d/neutronopenvswitch-agent.orig
sed -i 's,plugins/openvswitch/ovs_neutron_plugin.ini,plugin.ini,g' /etc/init.d/neutron-openvswitch-agen
t
```

配置 Nova 使用 Neutron 提供网络服务

```
openstack-config --set /etc/nova/nova.conf DEFAULT network_api_class nova.network.neutronv2.api.API
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_url http://controller0:9696
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_auth_strategy keystone
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_tenant_name service
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_username neutron
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_password neutron
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_admin_auth_url http://controller0:35357/v2.0
openstack-config --set /etc/nova/nova.conf DEFAULT linuxnet_interface_driver nova.network.linux_net.Lin
uxOVSInterfaceDriver
openstack-config --set /etc/nova/nova.conf DEFAULT firewall_driver nova.virt.firewall.NoopFirewallDrive
r
openstack-config --set /etc/nova/nova.conf DEFAULT security_group_api neutron

openstack-config --set /etc/nova/nova.conf DEFAULT service_neutron_metadata_proxy true
openstack-config --set /etc/nova/nova.conf DEFAULT neutron_metadata_proxy_shared_secret METADATA_SECRET
```

```
service openvswitch start
chkconfig openvswitch on
ovs-vsctl add-br br-int

service openstack-nova-compute restart
service neutron-openvswitch-agent start
chkconfig neutron-openvswitch-agent on
```

检查 agent 是否启动正常

```
neutron agent-list
```

8.4 创建初始网络

创建外部网络

```
neutron net-create ext-net --shared --router:external=True
```

为外部网络添加 subnet

```
neutron subnet-create ext-net --name ext-subnet \
--allocation-pool start=172.16.0.100,end=172.16.0.200 \
--disable-dhcp --gateway 172.16.0.1 172.16.0.0/24
```

创建住户网络,首先创建 demo 用户、租户已经分配角色关系

```
keystone user-create --name=demo --pass=demo --email=demo@example.com
keystone tenant-create --name=demo --description="Demo Tenant"
keystone user-role-add --user=demo --role=_member_ --tenant=demo
```

创建租户网络 demo-net

```
neutron net-create demo-net
```

为租户网络添加 subnet

```
neutron subnet-create demo-net --name demo-subnet --gateway 192.168.1.1 192.168.1.0/24
```

为租户网络创建路由，并连接到外部网络

```
neutron router-create demo-router
```

将 demo-net 连接到路由器

```
neutron router-interface-add demo-router $(neutron net-show demo-net|awk '/ subnets / { print $4 }')
```

网关

```
neutron router-gateway-set demo-router ext-net
```

启动一个 instance

```
nova boot --flavor m1.tiny --image $(nova image-list|awk '/ CirrOS / { print $2 }') --nic net-id=$(neut
ron net-list|awk '/ demo-net / { print $2 }') --security-group default demo-instance1
```

九.Dashboard 安装

安装 Dashboard 相关包

```
yum install memcached python-memcached mod_wsgi openstack-dashboard
```

配置 mencached

```
vi /etc/openstack-dashboard/local_settings

CACHES = {
'default': {
'BACKEND' : 'django.core.cache.backends.memcached.MemcachedCache',
'LOCATION' : '127.0.0.1:11211'
}
}
```

配置 Keystone hostname

```
vi /etc/openstack-dashboard/local_settings
OPENSTACK_HOST = "controller0"
```

启动 Dashboard 相关服务

```
service httpd start
```

```
service memcached start
chkconfig httpd on
chkconfig memcached on
```

打开浏览器验证,用户名：admin 密码：admin

```
http://10.20.0.10/dashboard
```