Centos7+Openstack juno(官方)安装手册

DCFabric对接

目录

[1.centos7 安装 3](#_Toc446509317)

[2.controller节点 5](#_Toc446509318)

[1.安装ntp服务 5](#_Toc446509319)

[2.安装mysql服务 5](#_Toc446509320)

[3 安装OpenStack 软件包 7](#_Toc446509321)

[4 OpenStack 存储库 7](#_Toc446509322)

[5 Messaging 服务 8](#_Toc446509323)

[6 添加 Identity 服务 9](#_Toc446509324)

[7 keystone安装和配置 11](#_Toc446509325)

[8 添加 Image 服务 15](#_Toc446509326)

[9 添加计算服务 18](#_Toc446509327)

[10 添加一个网络组件 25](#_Toc446509328)

[11 添加 Dashboard 30](#_Toc446509329)

[12 添加块存储服务 31](#_Toc446509330)

[13 安装libvirt补丁 37](#_Toc446509331)

[3.network节点 38](#_Toc446509332)

[1.安装ntp服务 38](#_Toc446509333)

[2 安装OpenStack 软件包 38](#_Toc446509334)

[3 OpenStack 存储库 39](#_Toc446509335)

[4 安装和配置网络节点 39](#_Toc446509336)

[4.compute节点 47](#_Toc446509337)

[1.安装ntp服务 47](#_Toc446509338)

[2 安装OpenStack 软件包 47](#_Toc446509339)

[3 OpenStack 存储库 48](#_Toc446509340)

[4 安装和配置计算程序组件 nova 48](#_Toc446509341)

[5 安装和配置Neutron计算节点 50](#_Toc446509342)

[6 安装libvirt补丁 55](#_Toc446509343)

[7 计算节点关闭防火墙 55](#_Toc446509344)

[5 对接DCFabric 56](#_Toc446509345)

[1 Openstack控制节点配置 56](#_Toc446509346)

[2 Openstack计算节点和网络节点配置 59](#_Toc446509347)

# 1.centos7 安装

修改网卡名称

vi /etc/sysconfig/grub

GRUB\_TIMEOUT=5

GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"

GRUB\_DEFAULT=saved

GRUB\_DISABLE\_SUBMENU=true

GRUB\_TERMINAL\_OUTPUT="console"

GRUB\_CMDLINE\_LINUX="rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-sun16 net.ifnames=0 biosdevname=0 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us rhgb quiet"

GRUB\_DISABLE\_RECOVERY="true"

以上红色字体为添加内容

3.执行命令

grub2-mkconfig -o /boot/grub2/grub.cfg

4.重启系统

reboot

5.修改 网卡配置

mv /etc/sysconfig/network-scripts/ifcfg-en0bs3 /etc/sysconfig/network-scripts/ifcfg-eth0

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

TYPE="Ethernet"

BOOTPROTO="dhcp"

DEFROUTE="yes"

IPV4\_FAILURE\_FATAL="no"

IPV6\_FAILURE\_FATAL="no"

NAME="eth1"

ONBOOT="yes"

PEERDNS="yes"

PEERROUTES="yes"

更改主机名（controller，network，computer）

hostnamectl set-hostname name

vi /etc/hosts

把3台机器的ip对应名称全部写入hosts

关闭防火墙：

systemctl stop firewalld.service

systemctl disable firewalld.service

# 2.controller节点

## 1.安装ntp服务

yum install ntp -y

1. 编辑/etc/ntp.conf 中的文件

server NTP\_SERVER iburst

restrict -4 default kod notrap nomodify

restrict -6 default kod notrap nomodify

用一个合适的更准确的主机名或 IP 地址的 NTP 服务器，替换 NTP\_SERVER。

1. 启动 NTP 服务，并将其配置为随系统自启动

systemctl enable ntpd.service

systemctl start ntpd.service

## 2.安装mysql服务

安装和配置数据库服务器

1. 安装包

yum install mariadb mariadb-server MySQL-python -y

2、编辑/etc/my.cnf 文件

A．在[mysqld]部分，设置 bind-address 的控制节点的管理网 IP 地址，以便通过 管理网络访问其他节点：

[mysqld] ...

bind-address = 10.0.0.11

B．在[mysqld]部分中，设置以下值启用有用的选项和 UTF-8 字符集：

[mysqld] ...

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

max\_connections=800

完成安装

1. 启动数据库服务，并将其配置为随系统自启动

systemctl enable mariadb.service

systemctl start mariadb.service

1. 固定数据库服务，包括选择合适的 root 帐户密码

mysql\_secure\_installation

## 3 安装OpenStack 软件包

请注意 禁用或移除任何自动更新服务，因为他们可能影响你的 OpenStack 环境。

在所有节点上执行这些过程。

1. 安装 yum-plugin-priorities 包 #

yum install yum-plugin-priorities -y

1. 安装 EPEL-release 组件

yum install http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-5.noarch.rpm -y

## 4 OpenStack 存储库

安装 RDO-RELEASE-JUNO 包

yum install http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno. rpm -y

完成安装

1. 升级你系统上的软件包

yum upgrade -y

请注意 如果在升级过程包括一个新的内核，重新启动您的系统将其激活。

1. RHEL 和 CentOS 默认启用了 SELinux。安装 SELinux 包自动管理 OpenStack 服务 的安全策略

yum install openstack-selinux -y

vi /etc/selinux/config  
SELINUX=disabled

setenforce 0

## 5 Messaging 服务

安装 RabbitMQ 消息代理服务

yum install rabbitmq-server -y

配置消息代理服务

1. 启动消息代理服务，并将其配置为自启动

systemctl enable rabbitmq-server.service

systemctl start rabbitmq-server.service

1. 运行下面的命令，用合适的密码替换 RABBIT\_PASS

rabbitmqctl change\_password guest guest

## 6 添加 Identity 服务

安装和配置

本节介绍如何在控制器节点，安装和配置 OpenStack 的身份服务。

配置 OpenStack 的身份服务，你必须创建一个数据库和一个管理令牌（token）。

1、创建数据库，请完成以下步骤

A．登陆数据库：

mysql -u root -p

B．运行脚本：

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'keystone';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'keystone';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'controller' IDENTIFIED BY 'keystone';

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'localhost' IDENTIFIED BY 'glance';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'%' IDENTIFIED BY 'glance';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance'@'controller' IDENTIFIED BY 'glance';

CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'localhost' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'%' IDENTIFIED BY 'nova';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova'@'controller' IDENTIFIED BY 'nova';

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'localhost' IDENTIFIED BY 'neutron';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'%' IDENTIFIED BY 'neutron';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron'@'controller' IDENTIFIED BY 'neutron';

CREATE DATABASE cinder;

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'localhost' IDENTIFIED BY 'cinder';

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'%' IDENTIFIED BY 'cinder';

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder'@'controller' IDENTIFIED BY 'cinder';

C．退出数据库

vi admin-openrc.sh

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=admin

export OS\_AUTH\_URL=http://controller:35357/v2.0

## 7 keystone安装和配置

1. 运行以下命令安装软件包

yum install <http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno.rpm> -y

yum install openstack-keystone python-keystoneclient –y

生成一个随机值作为初始配置过程中的管理令牌

openssl rand -hex 10

编辑/etc/keystone/keystone.conf 文件

A．在[DEFAULT]部分，定义初始管理的值：

[DEFAULT] ...

admin\_token = ADMIN\_TOKEN

使用您生成的随机值更换 ADMIN\_TOKEN

B．在[database]部分，配置数据库访问：

[database] ...

connection = mysql://keystone: KEYSTONE\_DBPASS@controller/keystone

用你选择的数据库密码替换 keystone\_dbpass。

C.（可选）在[DEFAULT]部分，启用详细日志记录协助解决问题：

[DEFAULT] ...

verbose = True

1. 创建通用的证书和密钥，限制访问相关文件

keystone-manage pki\_setup --keystone-user keystone --keystone-group keystone

chown -R keystone:keystone /var/log/keystone

chown -R keystone:keystone /etc/keystone/ssl

chmod -R o-rwx /etc/keystone/ssl

1. 填充 Keystone 身份服务数据库

su -s /bin/sh -c "keystone-manage db\_sync" keystone

4、启动身份服务和配置自启动

systemctl enable openstack-keystone.service

systemctl start openstack-keystone.service

5、默认情况下，身份服务存储在数据库中，无限过期的令牌积累大大增加了数 据库的大小，可能会降低服务的性能，特别是在资源有限的环境中。我们建议您 使用 cron 配置一个周期性任务，以清除过期的令牌：

(crontab -l -u keystone 2>&1 | grep -q token\_flush) || echo '@hourly /usr/bin/keystone-manage token\_flush >/var/log/keystone/ keystone-tokenflush.log 2>&1' >> /var/spool/cron/keystone

6、创建租户用户角色

vi .bashrc

export OS\_SERVICE\_TOKEN=ADMIN\_TOKEN

export OS\_SERVICE\_TOKEN=TOKEY\_KEY(replace with keystone.conf key)

export OS\_SERVICE\_ENDPOINT=http://192.168.51.118:35357/v2.0

source .bashrc

7、创建租户、用户和角色

A．创建 admin 租户：

keystone tenant-create --name admin --description "Admin Tenant"

B．创建 admin 用户：

keystone user-create --name admin --pass admin --email lyang@bnc.org.cn

C．创建 admin 角色：

keystone role-create --name admin

D．添加 admin 租户、admin 用户和 admin 角色：

keystone user-role-add --tenant admin --user admin --role admin

E．在默认情况下，dashboard 限制用户访问\_member\_作用域。

创建\_member\_作用域：

keystone role-create --name \_member\_

F．添加 admin 租户和用户到\_member\_角色：

keystone user-role-add --tenant admin --user admin --role \_member\_

OpenStack 服务还需要一个租户、用户和角色与其他服务进行交互。 •创建 service 租户：

keystone tenant-create --name service --description "Service Tenant"

创建 service entity 和 API 端点 1、身份服务管理你的 OpenStack 环境目录服务。服务使用此目录在您的环境中 查找其他服务 创建用于 Identity service 的服务：

keystone service-create --name keystone --type identity --description "OpenStack Identity"

创建用于 Identity service 的 API 端点

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

G．修改keystone\_endpoints\_basic.sh

HOST\_IP=192.168.51.118

EXT\_HOST\_IP=192.168.51.118

2个地址改成controller地址

# Keystone definitions

KEYSTONE\_REGION=regionOne

export SERVICE\_TOKEN=ADMIN\_TOKEN

export SERVICE\_TOKEN=f825ec015d8380c60f4d

export SERVICE\_ENDPOINT=http://controller:35357/v2.0

其中token key替换成keystone.conf内的key

bash ~/keystone\_endpoints\_basic.sh

source .bashrc

H、编辑 admin-openrc.sh 文件，添加以下内容

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=ADMIN\_PASS

export OS\_AUTH\_URL=http://controller:35357/v2.0

## 8 添加 Image 服务

安装和配置

本节介绍如何安装和配置 Image 服务，代号为 glance，在控制器节点上。为简单 起见，此存储镜像配置为本地文件系统。

配置的先决条件

在您安装和配置镜像服务之前，您必须创建一个数据库和身份服务终端凭据。

1、创建数据库，之前identity步骤中已完成这些步骤

2、执行 admin 凭证脚本文件 $ source admin-openrc.sh

3、创建身份服务凭据

A．创建 glance 用户：

keystone user-create --name glance --pass glance

B．将 glance 用户链接到 service 租户和 admin 角色：

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

C．创建 glance 服务：

keystone service-create --name glance --type image --description "OpenStack Image Service"

4、创建身份服务端点

keystone endpoint-create --service-id $(keystone service-list | awk '/ image / {print $2}') --publicurl http://controller:9292 --internalurl http://controller:9292 --adminurl http://controller:9292 --region regionOne

5、安装和配置 Image 服务组件

A、安装包

yum install openstack-glance python-glanceclient -y

B、编辑/etc/glance/glance-api.conf 文件

在[database]部分，

配置数据库访问：

[database] ...

connection = mysql://glance: GLANCE\_DBPASS@controller/glance

GLANCE\_DBPASS 替换为您选择的镜像服务数据库密码。

在[ keystone\_authtoken ]和[ paste\_deploy ]部分，

配置身份服务访问：

[keystone\_authtoken] ...

auth\_uri = <http://controller:5000/v2.0>

identity\_uri = <http://controller:35357>

admin\_tenant\_name = service

admin\_user = glance

admin\_password = GLANCE\_PASS

[paste\_deploy] ...

flavor = keystone

GLANCE\_PASS 替换为您选择的 glance 用户密码。

（可选）在[DEFAULT]部分，启用详细记录来帮助解决问题 [DEFAULT] ... verbose = True

C、编辑/etc/glance/glance-registry.conf 文件

在[database]部分，配置数据库访问：

[database] ...

connection = mysql://glance: GLANCE\_DBPASS@controller/glance

GLANCE\_DBPASS 替换为您选择的镜像服务数据库密码。

在[keystone\_authtoken]和[paste\_deploy]部分，配置身份服务访问： [keystone\_authtoken] ...

auth\_uri = <http://controller:5000/v2.0>

identity\_uri = <http://controller:35357>

admin\_tenant\_name = service

admin\_user = glance

admin\_password = GLANCE\_PASS

[paste\_deploy] ...

flavor = keystone

GLANCE\_PASS 替换为您选择的 glance 用户身份服务密码。

在[DEFAULT]部分，启用详细的日志记录，来协助排除潜在的故障：

[DEFAULT] ...

verbose = True

D、填充 Image 服务数据库

su -s /bin/sh -c "glance-manage db\_sync" glance

E、完成安装

启动镜像服务和配置随系统启动：

systemctl enable openstack-glance-api.service

systemctl enable openstack-glance-registry.service

systemctl start openstack-glance-api.service

systemctl start openstack-glance-registry.service

F、上传镜像

source admin-openrc.sh

glance image-create --name "cirros-0.3.4" --file cirros-0.3.4-x86\_64-disk.img --disk-format qcow2 --container-format bare --is-public True --progress

glance image-create --name "centos7" --file CentOS-7-x86\_64-GenericCloud-20150628\_01.qcow2 --disk-format qcow2 --container-format bare --is-public True --progress

G、测试

glance image-list

## 9 添加计算服务

安装和配置控制器节点

配置的先决条件

1. 创建数据库，之前identity已完成这些步骤
2. 执行 admin 凭证文件

source admin-openrc.sh

3、创建身份服务凭证，完成以下步骤

A．创建 nova 用户：

keystone user-create --name nova --pass nova

用一个合适的密码替换 NOVA\_PASS。

B．Nova 用户链接到 service 租户和 admin 角色：

keystone user-role-add --user nova --tenant service --role admin

C．创建 nova 服务：

keystone service-create --name nova --type compute --description "OpenStack Compute"

1. 创建计算服务端点

keystone endpoint-create --service-id $(keystone service-list | awk '/ compute / {print $2}') --publicurl http://controller:8774/v2/%\(tenant\_id\)s --internalurl http://controller:8774/v2/%\(tenant\_id\)s --adminurl http://controller:8774/v2/%\(tenant\_id\)s --region regionOne

1. 安装和配置计算控制器组件

A、安装包

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

B、编辑/etc/nova/nova.conf 文件，内容如下

[DEFAULT]

# LOGS/STATE

verbose=True

logdir=/var/log/nova

state\_path=/var/lib/nova

rootwrap\_config=/etc/nova/rootwrap.conf

verbose = True

# SCHEDULER

scheduler\_default\_filters=AggregateInstanceExtraSpecsFilter,AvailabilityZoneFilter,RamFilter,ComputeFilter

live\_migration\_bandwidth=0

live\_migration\_retry\_count=30

live\_migration\_flag=VIR\_MIGRATE\_UNDEFINE\_SOURCE,VIR\_MIGRATE\_PEER2PEER,VIR\_MIGRATE\_LIVE

# VOLUMES

volume\_api\_class=nova.volume.cinder.API

volume\_driver=nova.volume.driver.ISCSIDriver

volume\_group=cinder-volumes

volume\_name\_template=volume-%s

iscsi\_helper=tgtadm

rpc\_backend = rabbit

rabbit\_host = 192.168.51.118

rabbit\_password = guest

# COMPUTE

virt\_type = qemu

libvirt\_type=kvm

compute\_driver=libvirt.LibvirtDriver

instance\_name\_template=instance-%08x

api\_paste\_config=/etc/nova/api-paste.ini

# COMPUTE/APIS: if you have separate configs for separate services

# this flag is required for both nova-api and nova-compute

allow\_resize\_to\_same\_host=True

# APIS

osapi\_compute\_extension=nova.api.openstack.compute.contrib.standard\_extensions

enabled\_apis=ec2,osapi\_compute,metadata

# GLANCE

# NETWORK

network\_api\_class = nova.network.neutronv2.api.API

#neutron\_url = http://controller

#neutron\_auth\_strategy = keystone

#neutron\_admin\_tenant\_name = admin

#neutron\_admin\_username = admin

#neutron\_admin\_password = admin

#neutron\_admin\_auth\_url = http://controller:35357/v2.0

linuxnet\_interface\_driver = nova.network.linux\_net.LinuxOVSInterfaceDriver

firewall\_driver = nova.virt.firewall.NoopFirewallDriver

security\_group\_api = neutron

libvirt\_vif\_driver=nova.virt.libvirt.vif.LibvirtOpenVswitchDriver

vif\_plugging\_timeout = 10

vif\_plugging\_is\_fatal = False

service\_neutron\_metadata\_proxy = true

neutron\_metadata\_proxy\_shared\_secret =neutron

my\_ip= 192.168.51.118

#auto\_assign\_floating\_ip=True

#this base url must use real ip address

novncproxy\_base\_url=http://192.168.51.118:6080/vnc\_auto.html

# Change vncserver\_proxyclient\_address and vncserver\_listen to match each compute host

vncserver\_proxyclient\_address= 192.168.51.118

vncserver\_listen= 192.168.51.118

[neutron]

url = http://controller:9696

auth\_strategy = keystone

admin\_auth\_url = http://controller:35357/v2.0

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = neutron

[glance]

host = 192.168.51.118

# AUTHENTICATION

auth\_strategy=keystone

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = nova

admin\_password = nova

[database]

connection = mysql://nova:nova@controller /nova

[libvirt]

virt\_type = qemu

其中192.168.51.118均为controller地址

1. 填充 Nova 计算数据库

su -s /bin/sh -c "nova-manage db sync" nova

1. 完成安装 •启动计算服务和配置随系统自启动

systemctl enable openstack-nova-api.service

systemctl enable openstack-nova-cert.service

systemctl enable openstack-nova-consoleauth.service

systemctl enable openstack-nova-scheduler.service

systemctl enable openstack-nova-conductor.service

systemctl enable openstack-nova-novncproxy.service

systemctl restart openstack-nova-api.service

systemctl restart openstack-nova-cert.service

systemctl restart openstack-nova-consoleauth.service

systemctl restart openstack-nova-scheduler.service

systemctl restart openstack-nova-conductor.service

systemctl restart openstack-nova-novncproxy.service

## 10 添加一个网络组件

使用 OpenStack 网络(neutron)

安装和配置控制器节点 配置的先决条件

1. 创建数据库

此功能已在identity配置中完成

1. 执行 admin 凭证文件

source admin-openrc.sh

1. 创建身份服务凭证

A．创建 neutron 用户:

keystone user-create --name neutron --pass NEUTRON\_PASS

用一个合适的密码替换 NEUTRON\_PASS。

B．neutron 用户链接到 service 租户和 admin 角色：

keystone user-role-add --user neutron --tenant service --role admin

C．创建 neutron 服务：

keystone service-create --name neutron --type network --description "OpenStack Networking"

D．创建身份服务：

keystone endpoint-create --service-id $(keystone service-list | awk '/ network / {print $2}') --publicurl http://controller:9696 --adminurl http://controller:9696 --internalurl <http://controller:9696> --region regionOne

4．安装网络组件

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

5．配置网络服务器组件 •编辑/etc/neutron/neutron.conf 文件 内容如下

分别使用service租户(id)、nova 用户密码替换SERVICE\_TENANT\_ID 和NOVA\_PASS。

获取 service 租户(id)：

keystone tenant-get service

[DEFAULT]

core\_plugin = ml2

service\_plugins = router,neutron.services.loadbalancer.plugin.LoadBalancerPlugin

auth\_strategy = keystone

allow\_overlapping\_ips = True

agent\_down\_time = 120

notify\_nova\_on\_port\_status\_changes = True

notify\_nova\_on\_port\_data\_changes = True

nova\_url = http://controller:8774/v2

nova\_admin\_auth\_url = http://controller:35357/v2.0

nova\_region\_name = regionOne

nova\_admin\_username = nova

nova\_admin\_tenant\_id = bd2e4335c5a04136ad5f4b95be156e53

nova\_admin\_password = nova

rabbit\_host=192.168.51.118

rpc\_backend=rabbit

rabbit\_password = guest

verbose = True

[matchmaker\_redis]

[matchmaker\_ring]

[quotas]

[agent]

report\_interval = 30

[keystone\_authtoken]

auth\_uri = http://controller:5000/v2.0

identity\_uri = http://controller:35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = neutron

[database]

connection = mysql://neutron:neutron@controller:3306/neutron

[service\_providers]

service\_provider=LOADBALANCER:Haproxy:neutron.services.loadbalancer.drivers.haproxy.plugin\_driver.HaproxyOnHostPluginDriver:default

service\_provider=VPN:openswan:neutron.services.vpn.service\_drivers.ipsec.IPsecVPNDriver:default

6．配置模块2 层（ML2）插件

编辑/etc/neutron/plugins/ml2/ml2\_conf.ini 文件

1. 在[ML2]部分：

[ml2] ...

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

1. 在[ml2\_type\_gre]部分，配置隧道标识符(id)范围：

[ml2\_type\_gre] ...

tunnel\_id\_ranges = 1:1000

1. 在[securitygroup]部分：

[securitygroup] ...

enable\_security\_group = True

enable\_ipset = True

firewall\_driver = neutron.agent.linux.iptables\_firewall.OVSHybridIptablesFirewallDriver

7．完成安装

A．网络服务初始化脚本需要一个符号链接/etc/neutron/plugin.ini 插件配置文件 中，打开指向名为 ML2 层的 /etc/neutron/plugins/ml2/ml2\_conf.ini 文件。如果此 符号链接不存在，就使用以下命令创建它

ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

B．填充 neutron 数据库

su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade juno" neutron

8、重新启动计算服务

systemctl restart openstack-nova-api.service

systemctl restart openstack-nova-scheduler.service

systemctl restart openstack-nova-conductor.service

9、启动网络服务和配置随系统自启动

systemctl enable neutron-server.service

systemctl start neutron-server.service

## 11 添加 Dashboard

请注意 Dashboard 使用 VNC 客户端，浏览器必须支持 HTML5、Canvas 和 HTML5 WebSockets。 浏览器支持 noVNC 的详细信息, 请参阅 https://github.com/kanaka/noVNC/blob/master/README.md 和 https://github.com/kanaka/noVNC/。

安装和配置 本节描述如何安装和配置控制器节点上的 Dashboard。

安装 Dashboard 组件 •安装包：

yum install openstack-dashboard httpd mod\_wsgi memcached pythonmemcached -y

配置 dashboard •编辑/etc/openstack-dashboard/local\_settings 文件

1. 在 OpenStack 服务控制器节点上配置使用

dashboard：

OPENSTACK\_HOST = "192.168.51.118"

B．允许所有主机访问 dashboard： ALLOWED\_HOSTS = ['\*']

C．配置 memcached 会话存储服务：

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

请注意 注释掉其他会话存储配置。

D．(可选)配置时区： TIME\_ZONE = "TIME\_ZONE" 用一个合适的时区标识符替换 TIME\_ZONE。

完成安装 1、在 RHEL 和 CentOS、SELinux 上允许 web 服务器配置为连接到 OpenStack 服务

setsebool -P httpd\_can\_network\_connect on

2、由于包错误，dashboard CSS 无法正常加载。运行以下命令来解决这个问题

chown -R apache:apache /usr/share/openstack-dashboard/static

有关更多信息，请参见错误报告。

1. 启动 web 服务器和会话存储服务，配置随系统启动

systemctl enable httpd.service memcached.service

systemctl start httpd.service memcached.service

验证操作 本节描述如何验证操作 dashboard。

1、使用 web 浏览器访问 dashboard：<http://controller/dashboard>。

2、使用 admin 或 demo 用户验证凭证。

## 12 添加块存储服务

OpenStack 块存储 OpenStack 块存储服务(cinder)将持久性存储（你可以理解为移动硬盘，随取随用） 添加到一个虚拟机。块存储卷管理为 OpenStack 计算实例提供卷。服务还支持卷 快照管理、数量和类型。

安装和配置控制器节点 本节介绍如何安装和配置块存储服务，代号 cinder，控制器节点上。这个可选服 务至少需要一个额外的节点。

配置的先决条件 在安装和配置块存储服务之前，

您必须创建一个数据库和身份服务凭证包括端点。

1、创建数据库 此步骤identity步骤已完成

2、执行 admin 凭证文件

source admin-openrc.sh

3、创建身份服务凭证，完成以下步骤

A．创建一个 cinder 用户：

keystone user-create --name cinder --pass cinder

用一个合适的 cinder 用户密码替换 CINDER\_PASS。

1. cinder 用户链接到 service 租户和 admin 角色：

keystone user-role-add --user cinder --tenant service --role admin

1. 创建 cinder 服务：

keystone service-create --name cinder --type volume --description "OpenStack Block Storage"

请注意 块存储服务需要两个不同的服务支持 API 版本 1 和 2。

1. 创建块存储服务端点：

keystone endpoint-create --service-id $(keystone service-list | awk '/ volume / {print $2}') --publicurl http://controller:8776/v1/%\(tenant\_id\)s --internalurl http://controller:8776/v1/%\(tenant\_id\)s --adminurl http://controller:8776/v1/%\(tenant\_id\)s --region regionOne

keystone endpoint-create --service-id $(keystone service-list | awk '/ volumev2 / {print $2}') --publicurl http://controller:8776/v2/%\(tenant\_id\)s --internalurl http://controller:8776/v2/%\(tenant\_id\)s --adminurl http://controller:8776/v2/%\(tenant\_id\)s --region regionOne

请注意 块存储服务需要两个不同的端点支持 API 版本 1 和 2。

安装和配置块存储控制器组件

1. 安装包

yum install openstack-cinder python-cinderclient python-oslo-db -y

2、编辑/etc/cinder/cinder.conf 文件

A．在[database]部分，配置数据库访问：

[database] ...

connection = mysql://cinder: cinder@controller/cinder

CINDER\_DBPASS 替换为您选择的块存储数据库密码。

B．在[DEFAULT]部分，配置 RabbitMQ message broker 访问：

[DEFAULT] ...

rpc\_backend = rabbit

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

用您选择的 RABBIT 帐户密码替换 RABBIT\_PASS。

1. 在[DEFAULT]和[keystone\_authtoken]部分，配置身份服务访问：

[DEFAULT] ...

auth\_strategy = keystone

[keystone\_authtoken] ...

auth\_uri = <http://controller:5000/v2.0>

identity\_uri = <http://controller:35357>

admin\_tenant\_name = service

admin\_user = cinder

admin\_password = CINDER\_PASS

CINDER\_PASS 替换为您选择的 cinder 用户密码。

请注意 注释掉任何 auth\_host 、auth\_port、auth\_protocol 选项，因为 identity\_uri 选项 会替换他们。

D．在[DEFAULT]部分，my\_ip 选项配置为控制器节点的管理接口 IP 地址： [DEFAULT] ... my\_ip = 10.0.0.11

E．在[DEFAULT]部分，启用详细日志记录：

[DEFAULT] ...

verbose = True

1. 填充 cinder 数据库

su -s /bin/sh -c "cinder-manage db sync" cinder

完成安装 •启动块存储服务和配置随系统启动

systemctl enable openstack-cinder-api.service openstack-cinder-scheduler.service

systemctl start openstack-cinder-api.service openstack-cinder-scheduler.service

以下部分可选 /\*\*\*\*\*\*\*\*\*\*\*\*

配置一个块存储服务节点 控制器节点上配置服务后，配置一个块存储服务节点，其中包含服务卷的磁盘。 您可以配置 OpenStack 使用不同的存储系统（如 ceph、gfs、nfs 等）。这个例子 中，使用一个 LVM。

配置操作系统 控制器节点： •设置主机名 block1 和使用 10.0.0.41 管理网络接口 IP 地址。确保控制器节点的 IP 地址、主机名和块存储服务节点/etc/hosts 文件中列出每个系统。

使用“网络时间协议(NTP)”，从控制器节点保持时间同步。

创建一个逻辑卷 1、 安装 LVM 包 # yum install lvm2

请注意 默认一些发行版包括 LVM。

2、 启动 LVM 的元数据服务和配置随系统启动 # systemctl enable lvm2-lvmetad.service # systemctl start lvm2-lvmetad.service

3、创建 LVM 物理卷和卷组。本指南假定第二个磁盘/dev/sdb 被用于此目的 # pvcreate /dev/sdb # vgcreate cinder-volumes /dev/sdb

4、在/etc/lvm/lvm. conf 文件部分，添加 r/.\*/防止 LVM 扫描设备所使用的虚拟机 devices {

... filter = [ "a/sda1/", "a/sdb/", "r/.\*/"] ... }

请注意 在这个例子中，/dev/sda1 卷的操作系统为节点驻留，虽然/dev/sdb 留给 cinder-volumes。

安装和配置块存储服务节点组件 1、安装块存储服务 # yum install openstack-cinder targetcli python-oslo-db MySQL-python

2、编辑/etc/cinder/cinder.conf 文件 A．在[database]部分，配置数据库访问： [database] ... connection = mysql://cinder: CINDER\_DBPASS@controller/cinder

CINDER\_DBPASS 替换为您选择的块存储数据库密码。

B．在[DEFAULT]部分，配置 RabbitMQ message broker 访问： [DEFAULT] ... rpc\_backend = rabbit rabbit\_host = controller rabbit\_password = RABBIT\_PASS

用您选择的 RABBIT 账户密码替换 RABBIT\_PASS C．在[DEFAULT]和[keystone\_authtoken]部分，配置身份服务访问： [DEFAULT] ... auth\_strategy = keystone [keystone\_authtoken] ... auth\_uri = http://controller:5000/v2.0 identity\_uri = http://controller:35357 admin\_tenant\_name = service admin\_user = cinder admin\_password = CINDER\_PASS

CINDER\_PASS 替换为您选择的 cinder 用户密码。

请注意 注释掉任何 auth\_host、 auth\_port、auth\_protocol 选项，因为 identity\_uri 选项 将替换他们。

D．在[DEFAULT]部分，配置 my\_ip 选择： [DEFAULT] ... my\_ip = MANAGEMENT\_INTERFACE\_IP\_ADDRESS

用存储节点的管理网络接口 IP 地址替换 MANAGEMENT\_INTERFACE\_IP\_ADDRESS， 通常为 10.0.0.41 例子中的第一个体系结构的节点。

E．在[DEFAULT]部分，配置镜像的位置服务： [DEFAULT] ... glance\_host = controller

F．在[DEFAULT]部分，块存储配置为使用 iSCSI lioadm 服务： [DEFAULT] ... iscsi\_helper = lioadm

G．在[DEFAULT]部分，启用详细日志记录： [DEFAULT] ... verbose = True

完成安装 1、使用目标服务 # systemctl enable target.service

2、开始目标服务 # systemctl start target.service

3、启动和配置 cinder volume 服务随系统启动 # systemctl enable openstack-cinder-volume.service # systemctl start openstack-cinder-volume.service

验证操作 本节描述如何通过创建一个卷，验证操作块存储服务。

请注意 控制器节点上执行这些命令。

1、执行 demo 凭证文件 $ source demo-openrc.sh

2、创建一个 1GB 的卷 $ cinder create --display-name demo-volume1 1

3、验证创建的卷 $ cinder list

如果状态意味着不可用，则在/var/log/cinder 检查日志获取更多信息。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 13 安装libvirt补丁

yum install libvirt-daemon-driver-storage libvirt-daemon-config-nwfilter libvirt libvirt-client libvirt-daemon-driver-interface libvirt-daemon-driver-nodedev libvirt-daemon-driver-secret libvirt-daemon-config-network libvirt-daemon-driver-network libvirt-daemon-driver-lxc libvirt-daemon-kvm libvirt-daemon libvirt-daemon-driver-nwfilter libvirt-python libvirt-daemon-driver-qemu -y

控制节点重启 nova-conducter服务

systemctl restart openstack-nova-conductor

# 3.network节点

## 1.安装ntp服务

yum install ntp -y

1. 编辑/etc/ntp.conf 中的文件

server NTP\_SERVER iburst

用一个合适的更准确的主机名或 IP 地址的 NTP 服务器，替换 NTP\_SERVER。

1. 启动 NTP 服务，并将其配置为随系统自启动

# systemctl enable ntpd.service

# systemctl start ntpd.service

## 2 安装OpenStack 软件包

请注意 禁用或移除任何自动更新服务，因为他们可能影响你的 OpenStack 环境。

在所有节点上执行这些过程。

1. 安装 yum-plugin-priorities 包 #

yum install yum-plugin-priorities -y

1. 安装 EPEL-release 组件

yum install http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-5.noarch.rpm -y

## 3 OpenStack 存储库

安装 RDO-RELEASE-JUNO 包

yum install http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno. rpm -y

完成安装

1. 升级你系统上的软件包

yum upgrade -y

请注意 如果在升级过程包括一个新的内核，重新启动您的系统将其激活。

1. RHEL 和 CentOS 默认启用了 SELinux。安装 SELinux 包自动管理 OpenStack 服务 的安全策略

yum install openstack-selinux -y

vi /etc/selinux/config  
SELINUX=disabled

setenforce 0

## 4 安装和配置网络节点

网络节点主要处理内部和外部的路由以及虚拟网络的 DHCP 服务。

配置的先决条件 在安装和配置 OpenStack 网络之前，您必须配置特定的内核网络参数。 1、编辑/etc/sysctl.conf 文件，包含以下参数

net.ipv4.ip\_forward=1

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

1. 使配置生效

sysctl -p

1. 安装网络组件

yum install <http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno.rpm> -y

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

1. 配置网络常见的组件

常见的网络组件配置，包括身份验证机制、message broker、插件。

•编辑/etc/neutron/neutron.conf 文件，并完成以下步骤

A．在[database]部分，注释掉任何 connection 选项，因为网络节点不直接访问数 据库。

B．在[DEFAULT]部分，配置 RabbitMQ message broker 访问：

[DEFAULT] ... rpc\_backend = rabbit

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

用您选择的 RabbitMQ 账户密码替换 RABBIT\_PASS。

C．在[DEFAULT]和[keystone\_authtoken]部分，配置身份服务访问：

[DEFAULT] ...   
auth\_strategy = keystone

[keystone\_authtoken] ...

auth\_uri = <http://controller:5000/v2.0>

identity\_uri = <http://controller:35357>

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

NEUTRON\_PASS 替换为您选择的 neutron 用户密码。

请注意 注释掉任何 auth\_host 、auth\_port、auth\_protocol 选项，因为 identity\_uri 选项 将替换他们。

1. 在[DEFAULT]部分，在模块化 2 层中打开名为（ML2）的插件，使路由器服务 和 IP 地址重叠：

[DEFAULT] ...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

E．(可选)在[DEFAULT]部分，启用详细日志记录

[DEFAULT] ...

verbose = True

配置模块2 层（ML2）插件 ML2 插件使用开放 VSwitch（OVS）机制（agent），为实例（VM）建立虚拟网络 框架。

编辑/etc/neutron/plugins/ml2/ml2\_conf.ini 文件，并完成以下操作

在[ML2]部分：

[ml2] ...

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

B．在[ml2\_type\_flat]部分，配置外部网络：

[ml2\_type\_flat] ...

flat\_networks = external

C．在[ml2\_type\_gre]部分，配置隧道标识符(id)范围：

[ml2\_type\_gre] ...

tunnel\_id\_ranges = 1:1000

D．[securitygroup]部分：

[securitygroup] ...

enable\_security\_group = True

enable\_ipset = True

firewall\_driver = neutron.agent.linux.iptables\_firewall. OVSHybridIptablesFirewallDriver

1. 在(ovs)部分，配置 Open vSwitch(ovs)代理：

[ovs]

local\_ip = INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS

tunnel\_type = gre

enable\_tunneling = True

bridge\_mappings = external:br-ex

用网络节点上的隧道接口的实例 IP 地址替换 INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS。

F．配置第三层(L3)代理 第三层(L3)代理为虚拟网络提供路由服务。

编辑/etc/neutron/l3\_agent.ini 文件，并完成以下步骤 A．在[DEFAULT]部分，配置驱动程序和网络名称支持空间：

[DEFAULT] ...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver use\_namespaces = True

external\_network\_bridge = br-ex

(可选)在[DEFAULT]部分，启用详细日志记录：

[DEFAULT] ...

verbose = True

G．配置 DHCP 代理 DHCP 代理为虚拟网络提供 DHCP 服务。

编辑/etc/neutron/dhcp\_agent.ini 文件

在[DEFAULT]部分，配置驱动程序和启用名称空间：

[DEFAULT] ...

interface\_driver = neutron.agent.linux.interface.OVSInterfaceDriver dhcp\_driver= neutron.agent.linux.dhcp.Dnsmasq

use\_namespaces = True

B．(可选)在[DEFAULT]部分，启用详细日志记录：

[DEFAULT] ...

verbose = True

H．配置元数据代理 元数据代理提供配置信息，如凭证实例。

编辑/etc/neutron/metadata\_agent.ini 文件 A．在[DEFAULT]部分，配置访问参数：

[DEFAULT] ...

auth\_url = <http://controller:5000/v2.0>

auth\_region = regionOne

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

用您选择的 neutron 用户密码替换 NEUTRON\_PASS。

B．在[DEFAULT]部分，配置元数据：

[DEFAULT] ...

nova\_metadata\_ip = 192.168.53.90

C．在[DEFAULT]部分，配置元数据代理共享密钥：

[DEFAULT] ...

metadata\_proxy\_shared\_secret = METADATA\_SECRET

使用一个合适的元数据代理密码替换 METADATA\_SECRET。

D．在[DEFAULT]部分，启用详细日志记录：

[DEFAULT] ...

verbose = True

1. 在控制器节点上，编辑/etc/nova/nova.conf 文件

在[neutron]部分

[neutron] ...

service\_metadata\_proxy = True

metadata\_proxy\_shared\_secret = METADATA\_SECRET

使用你选择的元数据代理密码替换 METADATA\_SECRET。

J．在控制器节点上，重新启动计算 API 服务

systemctl restart openstack-nova-api.service

1. 配置 Open vSwitch(OVS)服务

启动 OVS 服务和配置随系统自启动

systemctl enable openvswitch.service

systemctl start openvswitch.service

添加外部桥接

ovs-vsctl add-br br-ex

添加一个端口到外部桥接，连接到物理外部网络 INTERFACE\_NAME 替换为实际的接口名称。例如 eth2 或 ens256。

ovs-vsctl add-port br-ex INTERFACE\_NAME

在你的环境中，暂时禁用客户外部网络接口 # ethtool -K INTERFACE\_NAME gro off

1. 完成安装

网络服务初始化脚本/etc/neutron/plugin.ini 插件配置文件，链接指向名为 ML2 /etc/neutron/plugins/ml2/ml2\_conf.ini 文件。如果此符号链接不存在，就创建它。

ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

由于包装错误，Open vSwitch 代理初始化脚本显式地看起来，配置文件 /etc/neutron/plugin.ini 不是一个符号链接，插件配置文件中打开指向名为 ML2。 运行以下命令来解决这个问题

cp /usr/lib/systemd/system/neutron-openvswitch-agent.service /usr/lib/systemd/system/neutron-openvswitch-agent.service.orig

sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' /usr/lib/systemd/system/neutron-openvswitch-agent.service

1. 启动网络服务和配置随系统启动

systemctl enable neutron-openvswitch-agent.service

systemctl enable neutron-l3-agent.service

systemctl enable neutron-dhcp-agent.service

systemctl enable neutron-metadata-agent.service

systemctl enable neutron-ovs-cleanup.service

systemctl start neutron-openvswitch-agent.service

systemctl start neutron-l3-agent.service

systemctl start neutron-dhcp-agent.service

systemctl start neutron-metadata-agent.service

N．验证操作 请注意 控制器节点上执行这些命令。

1、执行 admin 凭证文件

source admin-openrc.sh

2.查看 neutron 代理列表

neutron agent-list

# 4.compute节点

## 1.安装ntp服务

yum install ntp -y

1. 编辑/etc/ntp.conf 中的文件

server NTP\_SERVER iburst

用一个合适的更准确的主机名或 IP 地址的 NTP 服务器，替换 NTP\_SERVER。

1. 启动 NTP 服务，并将其配置为随系统自启动

systemctl enable ntpd.service

systemctl start ntpd.service

## 2 安装OpenStack 软件包

请注意 禁用或移除任何自动更新服务，因为他们可能影响你的 OpenStack 环境。

在所有节点上执行这些过程。

1. 安装 yum-plugin-priorities 包 #

yum install yum-plugin-priorities -y

1. 安装 EPEL-release 组件

yum install http://dl.fedoraproject.org/pub/epel/7/x86\_64/e/epel-release-7-5.noarch.rpm -y

## 3 OpenStack 存储库

安装 RDO-RELEASE-JUNO 包

yum install http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno. rpm -y

完成安装

1. 升级你系统上的软件包

yum upgrade -y

请注意 如果在升级过程包括一个新的内核，重新启动您的系统将其激活。

1. RHEL 和 CentOS 默认启用了 SELinux。安装 SELinux 包自动管理 OpenStack 服务 的安全策略

yum install openstack-selinux -y

vi /etc/selinux/config  
SELINUX=disabled

setenforce 0

## 4 安装和配置计算程序组件 nova

1. 安装包

yum install <http://rdo.fedorapeople.org/openstack-juno/rdo-release-juno.rpm> -y

yum install openstack-nova-compute -y

1. 编辑/etc/nova/nova.conf 文件，内容等同于controller节点的nova.conf

其中修改：

my\_ip=compute节点地址

vncserver\_proxyclient\_address=compute节点地址

vncserver\_listen= compute节点地址

1. 完成安装 1、确定您的计算节点支持虚拟机硬件加速

egrep -c '(vmx|svm)' /proc/cpuinfo

如果该命令返回一个值或更多，则你的计算节点支持硬件加速，通常不需要额外 的配置。 相反，如果该命令返回零个值，则计算节点不支持硬件加速，你必须配置 libvirt KVM 使用 QEMU。

在/etc/nova/nova.conf 文件，编辑[libvirt]部分如下： [libvirt] ... virt\_type = qemu

1. 启动计算服务并配置随系统自启动

systemctl enable libvirtd.service

systemctl start libvirtd.service

systemctl enable openstack-nova-compute.service

systemctl start openstack-nova-compute.service

## 5 安装和配置Neutron计算节点

计算节点处理连接和安全组的实例。

配置先决条件 在安装和配置 OpenStack 网络之前，您必须配置特定的内核网络参数。

1. 编辑/etc/sysctl.conf 文件包含以下参数

net.ipv4.conf.all.rp\_filter=0

net.ipv4.conf.default.rp\_filter=0

使配置生效

sysctl -p

安装网络组件

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

配置网络常见的组件

常见的网络组件配置包括身份验证机制、message broker、插件。

编辑/etc/neutron/neutron.conf 文件

A． 在[database] 部分，注释掉任何 connection 选项，因为计算节点不直接访问 数据库。

B．在[DEFAULT]部分，配置 RabbitMQ message broker 访问：

[DEFAULT] ... rpc\_backend = rabbit

rabbit\_host = controller

rabbit\_password = RABBIT\_PASS

用您选择的 RabbitMQ 帐户密码替换 RABBIT\_PASS。

C．在[DEFAULT]和[keystone\_authtoken]部分，配置身份服务访问：

[DEFAULT] ...

auth\_strategy = keystone

[keystone\_authtoken] ...

auth\_uri = <http://controller:5000/v2.0>

identity\_uri = <http://controller:35357>

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = NEUTRON\_PASS

用您选择的 neutron 用户密码替换 NEUTRON\_PASS。

请注意 注释掉任何 auth\_host 、auth\_port、auth\_protocol 选项，因为 identity\_uri 选项 将替换他们。

D．在[DEFAULT]部分，模块化 2 层中打开名为（ML2）插件，使路由器服务和 IP 地址重叠：

[DEFAULT] ...

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

E．（可选）

[DEFAULT] ...

verbose = True

配置模块2 层（ML2）插件 插件（ML2）使用 Open vSwitch 中打开的名为 ML2(OVS)机制(agent)来构建虚拟网 络的框架实例。

•编辑/etc/neutron/plugins/ml2/ml2\_conf.ini 文件

1. 在[ml2]部分：

[ml2] ...

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

1. 在[ml2\_type\_gre]部分，配置隧道标识符(id)范围：

[ml2\_type\_gre] ...

tunnel\_id\_ranges = 1:1000

1. 在[securitygroup]部分：

[securitygroup] ...

enable\_security\_group = True

enable\_ipset = True

firewall\_driver = neutron.agent.linux.iptables\_firewall. OVSHybridIptablesFirewallDriver

1. 在(ovs)部分，配置 Open vSwitch(ov)代理：

[ovs]

local\_ip = INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS

tunnel\_type = gre

enable\_tunneling = True

用计算节点上的实例隧道网络接口 IP 地址替换 INSTANCE\_TUNNELS\_INTERFACE\_IP\_ADDRESS。

配置 Open vSwitch(ov)服务 •启动 OVS 服务和配置随机启动

systemctl enable openvswitch.service

systemctl start openvswitch.service

使用网络配置计算 •编辑/etc/nova/nova.conf 文件

1. 在[DEFAULT]部分：

[DEFAULT] ...

network\_api\_class = nova.network.neutronv2.api.API

security\_group\_api = neutron

linuxnet\_interface\_driver = nova.network.linux\_net.LinuxOVSInterfaceDriver firewall\_driver = nova.virt.firewall.NoopFirewallDriver

1. 在[neutron]部分，配置访问参数：

[neutron] ...

url = <http://controller:9696>

auth\_strategy = keystone

admin\_auth\_url = <http://controller:35357/v2.0>

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = NEUTRON\_PASS

用您选择的 neutron 用户身份服务密码替换 NEUTRON\_PASS。

完成安装

1. 做软连接

ln -s /etc/neutron/plugins/ml2/ml2\_conf.ini /etc/neutron/plugin.ini

由于包装错误，Open vSwitch 代理初始化脚本显式地看起来 Open vSwitch 插件配 置文件不是一个符号链接，/etc/neutron/plugin.ini 插件配置文件中打开指向名为 ML2。运行以下命令来解决这个问题

cp /usr/lib/systemd/system/neutron-openvswitch-agent.service /usr/lib/systemd/system/neutron-openvswitch-agent.service.orig

sed -i 's,plugins/openvswitch/ovs\_neutron\_plugin.ini,plugin.ini,g' /usr/lib/systemd/system/neutron-openvswitch-agent.service

1. 重启计算服务

systemctl restart openstack-nova-compute.service

1. 启动 Open vSwitch(ovs)代理和配置随系统启动

systemctl enable neutron-openvswitch-agent.service

systemctl start neutron-openvswitch-agent.service

验证操作 请注意 控制器节点上执行这些命令

1. 执行 admin 凭证文件

source admin-openrc.sh

1. 查看 neutron 代理列表

neutron agent-list

## 6 安装libvirt补丁

yum install libvirt-daemon-driver-storage libvirt-daemon-config-nwfilter libvirt libvirt-client libvirt-daemon-driver-interface libvirt-daemon-driver-nodedev libvirt-daemon-driver-secret libvirt-daemon-config-network libvirt-daemon-driver-network libvirt-daemon-driver-lxc libvirt-daemon-kvm libvirt-daemon libvirt-daemon-driver-nwfilter libvirt-python libvirt-daemon-driver-qemu –y

计算节点重启

systemctl restart libvirtd

systemctl restart openstack-nova-compute

systemctl restart neutron-openvswitch-agent

## 7 计算节点关闭防火墙

systemctl stop firewalld.service

systemctl disable firewalld.service

# 5 对接DCFabric

### 1 Openstack控制节点配置

1. 工具脚本获取

在“tools/openstack-tools”目录下有shell脚本“gnflush-controller.sh”和ml2驱动脚本“mechanism\_gnflush.py”用于设置计算节点和网络节点与数据中心SDN控制器对接的配置。

将这个脚本拷贝到Openstack控制节点服务器上如下目录下：

|  |
| --- |
| /usr/lib/python2.7/site-packages/neutron/plugins/ml2/drivers/ |

1. 关闭相关服务

在Openstack控制中心服务器上执行如下命令停止相关服务：

|  |
| --- |
| systemctl stop neutron-server  systemctl stop neutron-openvswitch-agent  systemctl disable neutron-openvswitch-agent |

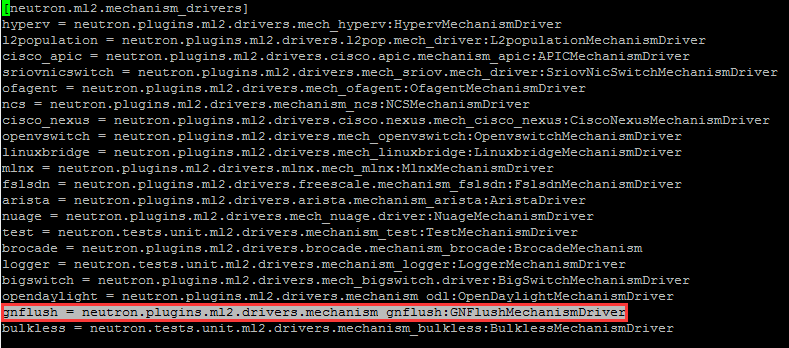
1. 配置ml2 plugin

在数据中心文件“entry\_points.txt”中“[neutron.ml2.mechanism\_drivers]”配置节点下增加GNFlush控制器ML2 plugin的entry point：

|  |
| --- |
| vi /usr/lib/python2.7/site-packages/neutron-2014.2.3-py2.7.egg-info/entry\_points.txt |

参考下图添加如下内容：

|  |
| --- |
| gnflush = neutron.plugins.ml2.drivers.mechanism\_gnflush:GNFlushMechanismDriver |



1. 配置ml2

配置ml2前，需要准备如下信息：

* 租户网络类型：管理员可以根据需要选择“gre”或者“vlan”。
* 控制器IP地址：数据中心SDN控制器所在服务器的IP地址。
* 控制器Rest服务端口：Rest服务监听端口需要查看SDN控制器配置文件中“[rest\_port]”配置，默认为“8081”

在Openstack控制中心服务器上执行如下命令修改ml2配置，以租户网络类型为“gre”为例：

|  |
| --- |
| crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers gnflush |

在“/etc/neutron/plugins/ml2/ml2\_conf.ini”文件最后面添加“ml2\_gnflush”的选项:

|  |
| --- |
| [ml2\_gnflush]  password = admin  username = admin  url = http://<控制器IP>:<控制Rest服务端口>/gn/neutron |

1. 创建ml2数据库

在Openstack控制中心服务器上执行如下命令：

|  |
| --- |
| mysql -u root -p  drop database if exists neutron\_ml2;  create database neutron\_ml2 character set utf8;  grant all on neutron\_ml2.\* to 'neutron'@'%';  grant all on neutron\_ml2.\* to 'neutron'@'controller' IDENTIFIED BY 'neutron';  grant all on neutron\_ml2.\* to 'neutron'@'localhost' IDENTIFIED BY 'neutron';  neutron-db-manage --config-file /usr/share/neutron/neutron-dist.conf --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini upgrade head |

1. 启动相关服务

在Openstack控制中心服务器上执行如下命令：

|  |
| --- |
| systemctl start neutron-server  systemctl disable neutron-linuxbridge-agent  systemctl stop neutron-linuxbridge-agent |

### 2 Openstack计算节点和网络节点配置

1. 工具脚本获取

在“tools/openstack-tools”目录下有shell脚本“gnflush-controller.sh”和ml2驱动脚本“mechanism\_gnflush.py”用于设置计算节点和网络节点与数据中心SDN控制器对接的配置。

将“mechanism\_gnflush.py”这个脚本拷贝到Openstack控制节点服务器上如下目录下：

|  |
| --- |
| /usr/lib/python2.7/site-packages/neutron/plugins/ml2/drivers/ |

将“gnflush-controller.sh”拷贝到服务器上任意目录后续使用。

1. 关闭相关服务

在所有Openstack计算节点和网络节点上服务器上执行如下命令：

|  |
| --- |
| systemctl disable neutron-linuxbridge-agent  systemctl stop neutron-linuxbridge-agent |

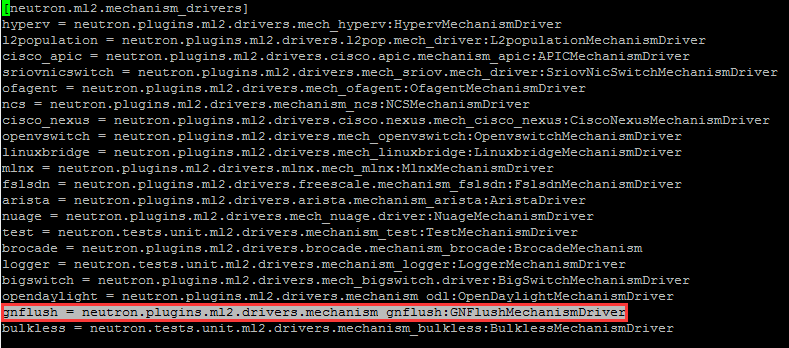
1. 配置ml2 plugin

在数据中心文件“entry\_points.txt”中“[neutron.ml2.mechanism\_drivers]”配置节点下增加GNFlush控制器ML2 plugin的entry point：

|  |
| --- |
| vi /usr/lib/python2.7/site-packages/neutron-2014.2.3-py2.7.egg-info/entry\_points.txt |

参考下图添加如下内容：

|  |
| --- |
| gnflush = neutron.plugins.ml2.drivers.mechanism\_gnflush:GNFlushMechanismDriver |



1. 配置ml2

配置ml2前，需要准备如下信息：

* 租户网络类型：管理员可以根据需要选择“gre”或者“vlan”。
* 控制器IP地址：数据中心SDN控制器所在服务器的IP地址。
* 控制器Rest服务端口：Rest服务监听端口需要查看SDN控制器配置文件中“[rest\_port]”配置，默认为“8081”

在Openstack控制中心服务器上执行如下命令修改ml2配置，以租户网络类型为“gre”为例：

|  |
| --- |
| crudini --set /etc/neutron/plugins/ml2/ml2\_conf.ini ml2 mechanism\_drivers gnflush |

在“/etc/neutron/plugins/ml2/ml2\_conf.ini”文件最后面添加“ml2\_gnflush”的选项:

|  |
| --- |
| [ml2\_gnflush]  password = admin  username = admin  url = http://192.168.53.3:8081/gn/neutron |

1. 配置Openvswitch

配置前，需要准备如下信息：

* 本地IP：当前服务器的IP地址。
* 网卡名称：用于建立隧道的物理网卡。
* 控制器IP地址：数据中心SDN控制器所在服务器的IP地址。

在所有Openstack计算节点和网络节点上服务器上执行如下命令，可以通过“--help”选项查看帮助信息：

|  |
| --- |
| 网络节点命令如下  sh gnflush-controller.sh --local\_ip 192.168.51.118 --provider\_mappings eth0 --gnflush\_ip 192.168.53.71 --external\_provider br-ex  计算节点命令如下  sh gnflush-controller.sh --local\_ip 192.168.53.52 --provider\_mappings eth1 --gnflush\_ip 192.168.53.3  ovs-vsctl add-br br-int  ovs-vsctl set-controller br-int tcp:192.168.53.70:6633  ovs-vsctl add-port br-int eth0 |

1. 启动相关服务

上执行如下命令启动服务：

|  |
| --- |
|  |

# 6 其他命令

安装iperf

yum install <http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm>

yum install iperf