Recommend all nodes install Centos 7.0

The openstack version is juno

Simple exmaple: one controller node, one compute node, one network node

**Configure Networking**

Static IP is recommended. Only use one public interface to communicate among three nodes. vim /etc/sysconfig/network-scripts/ifcfg-$YourNIC

Controller node IP address: 10.124.21.143 compute node IP address:10.124.21.144 network node IP address: 10.124.21.145

Network mask:255.255.255.0(or /24)

Default gateway:10.124.21.1

Network:10.124.21.0

**Configure name resolution**

Edit /etc/hosts file to contain the following items:

127.0.0.1 localhost

#controller node

10.124.21.143 Juno-Controller

#compute node

10.124.21.144 Juno-Compute

#network node

10.124.21.145 Juno-Network

**Verify connectivity**

verify network connectivity to the Internet and among the nodes is recommended. BTW, you need configure cisco DNS server for connectivity to the internet.

Cisco nameserver :64.104.123.144

You can use ping command to verify the connectivity to the internet and among the nodes.

For example: controller node-> ping –c 4 cisco.com  ping –c 4 Juno-Compute ping –c 4 Juno-Network

**Configure NTP**

1. controller node

yum install ntp

edit /etc/ntp.conf file :

server 10.64.58.50

restrict -4 default kod notrap nomodify

restrict -6 default kod notrap nomodify

systemctl enable ntpd.service

systemctl start ntpd.service

1. other nodes

yum install ntp

edit /etc/ntp.conf file :

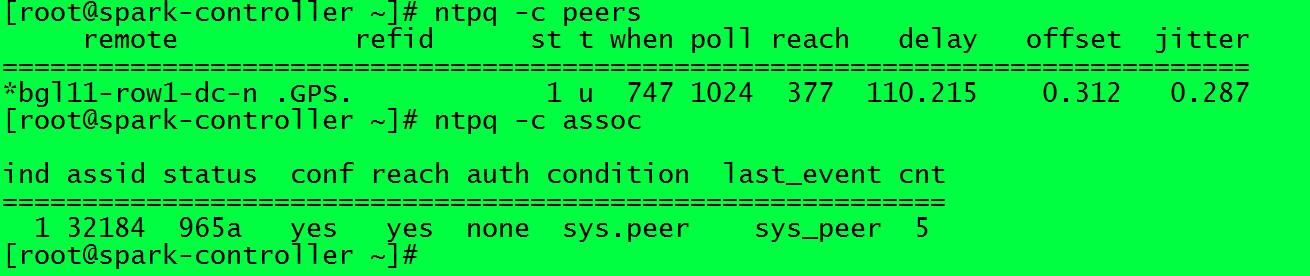
server Juno-Controller iburst

systemctl enable ntpd.service

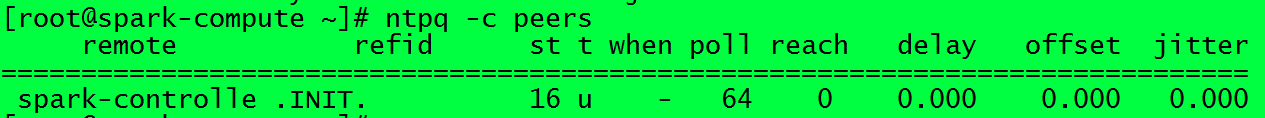
systemctl start ntpd.service

1. verify operation

Run the commands on controller node:



Run the commands on other nodes:



**Openstack packages**

All nodes:

yum install yum-plugin-priorities

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

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

yum upgrade

yum install openstack-selinux

**database**

controller node:

yum install mariadb mariadb-server MySQL-python

Edit the /etc/my.cnf file and complete the following actions:

[mysqld]

...

bind-address = 10.124.21.143

default-storage-engine = innodb

innodb\_file\_per\_table

collation-server = utf8\_general\_ci

init-connect = 'SET NAMES utf8'

character-set-server = utf8

systemctl enable mariadb.service

systemctl start mariadb.service

mysql\_secure\_installation

systemctl stop firewalld.service

systemctl disable firewalld.service

sed -i 's/enforcing/disabled/g' /etc/selinux/config

echo 0 > /sys/fs/selinux/enforce

**Messaging server**

controller node:

yum install rabbitmq-server

systemctl enable rabbitmq-server.service

systemctl start rabbitmq-server.service

rabbitmqctl change\_password guest cisco123

for RabbitMQ version 3.3.0 or newer,  you must enable remote access for the guest account:

rabbitmqctl status | grep rabbit

Status of node 'rabbit@spark-controller' ...

{running\_applications,[{rabbit,"RabbitMQ","3.3.5"},

edit the /etc/rabbitmq/rabbitmq.config file and configure loopback\_users to reference an empty list:

[{rabbit, [{loopback\_users, []}]}].

systemctl restart rabbitmq-server.service

**Keystone Installation**

**controller node:**

1. mysql -uroot -pcisco123

grant all privileges on \*.\* to root@"%" identified by "cisco123" with grant option;

FLUSH PRIVILEGES;

use mysql;

delete from user where user="";

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone\_admin'@'%' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone\_admin'@'localhost' IDENTIFIED BY 'cisco123';

yum install openstack-keystone python-keystoneclient

1. Edit the /etc/keystone/keystone.conf file and complete the following actions:

[DEFAULT]

admin\_token = cisco123

verbose = True

[database]

connection = mysql://keystone\_admin:cisco123@10.124.21.143/keystone

[token]

provider = keystone.token.providers.uuid.Provider

driver = keystone.token.persistence.backends.sql.Token

[revoke]

driver = keystone.contrib.revoke.backends.sql.Revoke

1. Create generic certificates and keys and restrict access to the associated files:

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

echo $? to Check whether the command is executed successfully

check whether the user keystone\_admin can access database

mysql -h10.124.21.143 -ukeystone\_admin -pcisco123 keystone

1. systemctl enable openstack-keystone.service

systemctl start openstack-keystone.service

1. use cron to configure a periodic task that purges expired tokens hourly

(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

**Create tenants, users, and roles**

1. configure prerequisites

export OS\_SERVICE\_TOKEN=cisco123

export OS\_SERVICE\_ENDPOINT=http://10.124.21.143:35357/v2.0

unset http\_proxy https\_proxy(if you set proxy before)

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

keystone user-create --name admin --pass cisco123 --email [xinrwu@cisco.com](mailto:xinrwu@cisco.com)

keystone role-create --name admin

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

keystone tenant-create --name demo --description "Demo Tenant"

keystone user-create --name demo --tenant demo --pass cisco123 --email [xinrwu@cisco.com](mailto:xinrwu@cisco.com)

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

**Create the service entity and API endpoint**

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

keystone endpoint-create --service-id $(keystone service-list | awk '/ identity / {print $2}') --publicurl http://10.124.21.143:5000/v2.0 --internalurl http://10.124.21.143:5000/v2.0 --adminurl http://10.124.21.143:35357/v2.0 --region regionOne

**Verify operation**

1. unset OS\_SERVICE\_TOKEN OS\_SERVICE\_ENDPOINT
2. As the admin tenant and user, request an authentication token:

keystone --os-tenant-name admin --os-username admin --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 token-get

1. As the admin tenant and user, list tenants to verify that the admin tenant and user can execute admin-only CLI commands and that the Identity service contains the tenants that created before:

keystone --os-tenant-name admin --os-username admin --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 tenant-list

1. As the admin tenant and user, list users to verify that the Identity service contains the users that created before:

keystone --os-tenant-name admin --os-username admin --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 user-list

1. As the admin tenant and user, list roles to verify that the Identity service contains the role that created before:

keystone --os-tenant-name admin --os-username admin --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 role-list

1. As the demo tenant and user, request an authentication token:

keystone --os-tenant-name demo --os-username demo --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 token-get

1. As the demo tenant and user, attempt to list users to verify that you cannot execute admin-only CLI commands:

keystone --os-tenant-name demo --os-username demo --os-password cisco123 --os-auth-url http://10.124.21.143:35357/v2.0 user-list

You are not authorized to perform the requested action: admin\_required (HTTP 403)

**create the scripts**

Edit the admin-openrc.sh file and add the following content:

export OS\_TENANT\_NAME=admin

export OS\_USERNAME=admin

export OS\_PASSWORD=cisco123

export OS\_AUTH\_URL=http://10.124.21.143:35357/v2.0

export OS\_SERVICE\_ENDPOINT=http://10.124.21.143:35357/v2.0

export OS\_SERVICE\_TOKEN=cisco123

export OS\_AUTH\_STRATEGY=keystone

To run clients as a certain tenant and user, you can simply load the associated client environment script prior to running them. For example, to load the location of the Identity service and admin tenant and user credentials:

source admin-openrc.sh

**Glance Installation**

**Controller node:**

1. mysql -uroot -pcisco123

CREATE DATABASE glance;

GRANT ALL PRIVILEGES ON glance.\* TO 'glance\_admin'@'localhost' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON glance.\* TO 'glance\_admin'@'%' IDENTIFIED BY 'cisco123';

source admin-openrc.sh

keystone user-create --name glance --pass cisco123

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

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

keystone endpoint-create --service-id $(keystone service-list | awk '/ image / {print $2}') --publicurl http://10.124.21.143:9292 --internalurl http://10.124.21.143:9292 --adminurl http://10.124.21.143:9292 --region regionOne

1. yum install openstack-glance python-glanceclient

Edit the /etc/glance/glance-api.conf  file and  /etc/glance/glance-registry.conf file,then complete the following actions:

[database]

connection = mysql://glance\_admin:cisco123@10.124.21.143/glance

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = glance

admin\_password = cisco123

[paste\_deploy]

flavor = keystone

[glance\_store]

default\_store = file

filesystem\_store\_datadir = /var/lib/glance/images/

[DEFAULT]

notification\_driver = noop

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

systemctl enable openstack-glance-api.service openstack-glance-registry.service

systemctl start openstack-glance-api.service openstack-glance-registry.service

## Verify operation

mkdir /tmp/images

yum install wget

wget -P /tmp/images [http://cdn.download.cirros-cloud.net/0.3.3/](http://cdn.download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img)[cirros-0.3.3-x86\_64-disk.img](http://cdn.download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img)

source admin-openrc.sh

glance image-create --name "cirros-0.3.3-x86\_64" --file /tmp/images/cirros-0.3.3-x86\_64-disk.img --disk-format qcow2 --container-format bare --is-public True --progress

glance image-list

rm -rf /tmp/images

**Nova Installation**

**Controller node:**

1. mysql -uroot -pcisco123

CREATE DATABASE nova;

GRANT ALL PRIVILEGES ON nova.\* TO 'nova\_admin'@'localhost' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON nova.\* TO 'nova\_admin'@'%' IDENTIFIED BY 'cisco123';

keystone user-create --name nova --pass cisco123

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

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

keystone endpoint-create --service-id $(keystone service-list | awk '/ compute / {print $2}') --publicurl http://10.124.21.143:8774/v2/%\(tenant\_id\)s --internalurl http://10.124.21.143:8774/v2/%\(tenant\_id\)s --adminurl http://10.124.21.143:8774/v2/%\(tenant\_id\)s --region regionOne

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

Edit the /etc/nova/nova.conf file and complete the following actions:

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

verbose = True

vncserver\_listen = 10.124.21.143

vncserver\_proxyclient\_address = 10.124.21.143

my\_ip=10.124.21.143

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = nova

admin\_password = cisco123

[glance]

host = 10.124.21.143

[database]

connection = mysql://nova\_admin:cisco123@10.124.21.143/nova

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

systemctl enable openstack-nova-api.service openstack-nova-cert.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service

systemctl start openstack-nova-api.service openstack-nova-cert.service openstack-nova-consoleauth.service openstack-nova-scheduler.service openstack-nova-conductor.service openstack-nova-novncproxy.service

nova image-list

**compute node:**

1. yum install openstack-nova-compute sysfsutils

Edit the /etc/nova/nova.conf file and complete the following actions:

[glance]

host = 10.124.21.143

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = nova

admin\_password = cisco123

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

my\_ip=10.124.21.144

vnc\_enabled = True

vncserver\_listen = 0.0.0.0

vncserver\_proxyclient\_address = 10.124.21.144

novncproxy\_base\_url = http://10.124.21.143:6080/vnc\_auto.html

verbose = True

#compute\_driver=nova.virt.libvirt.LibvirtDriver

compute\_driver=libvirt.LibvirtDriver

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

if this command returns a value of zero, your compute node does not support hardware acceleration and you must configure libvirt to use QEMU instead of KVM.

Edit the [libvirt] section in the /etc/nova/nova.conf file as follows:

[libvirt]

virt\_type = qemu

systemctl enable libvirtd.service openstack-nova-compute.service

systemctl start libvirtd.service openstack-nova-compute.service

!!!IMPORTANT: on controller node:(mariadb remote connection)

iptables -I INPUT -p udp --dport 3306 -j ACCEPT

iptables -I INPUT -p tcp --dport 3306 --syn -j ACCEPT

iptables-save

service mariadb restart

on controller node:(rabbitmq remote connection)

iptables -I INPUT -p udp --dport 5672 -j ACCEPT

iptables -I INPUT -p tcp --dport 5672 --syn -j ACCEPT

iptables-save

systemctl restart rabbitmq-server.service

then, you can check mariadb remote connection:

mysql -h10.124.21.143 -unova\_admin -pcisco123 nova

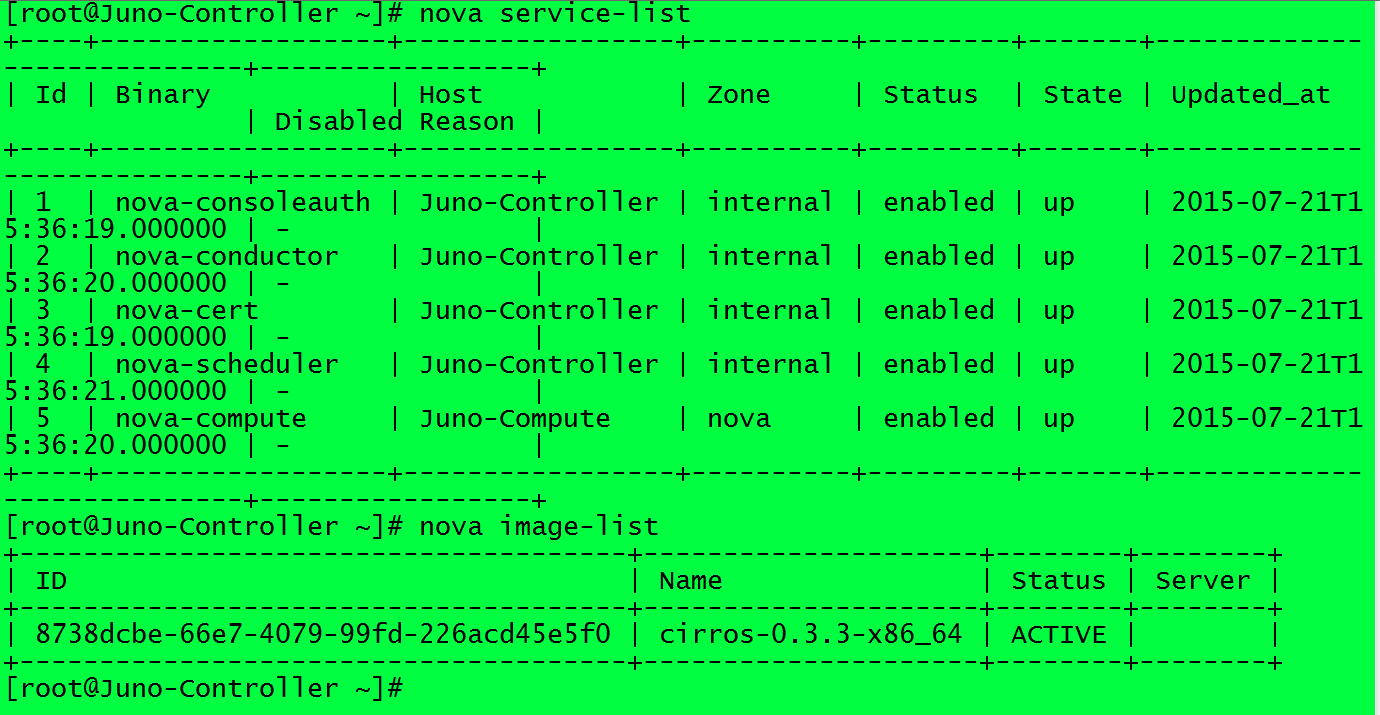
rabbitmq remote connection:

telnet 10.124.21.143 5672

1. verify operation

on controller node:

source admin-openrc.sh



**Neutron Installation**

**Controller node:**

1. mysql -uroot -pcisco123

CREATE DATABASE neutron;

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron\_admin'@'localhost' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON neutron.\* TO 'neutron\_admin'@'%' IDENTIFIED BY 'cisco123';

keystone user-create --name neutron --pass cisco123

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://10.124.21.143:9696 --adminurl http://10.124.21.143:9696 --internalurl http://10.124.21.143:9696 --region regionOne

1. yum install openstack-neutron openstack-neutron-ml2 python-neutronclient which

Edit the /etc/neutron/neutron.conf file and complete the following actions:

[database]

connection = mysql://neutron\_admin:cisco123@10.124.21.143/neutron

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = cisco123

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

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

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

nova\_url = http://10.124.21.143:8774/v2

nova\_admin\_auth\_url = http://10.124.21.143:35357/v2.0

nova\_region\_name = regionOne

nova\_admin\_username = nova

nova\_admin\_tenant\_id = $ *SERVICE\_TENANT\_ID*

nova\_admin\_password = cisco123

verbose = True

To obtain the service tenant identifier (id):

keystone tenant-get service

Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file and complete the following actions:

[ml2]

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

Edit the /etc/nova/nova.conf file on the controller node and complete the following actions:

[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

[neutron]

url = http://10.124.21.143:9696

auth\_strategy = keystone

admin\_auth\_url = http://10.124.21.143:35357/v2.0

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = cisco123

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

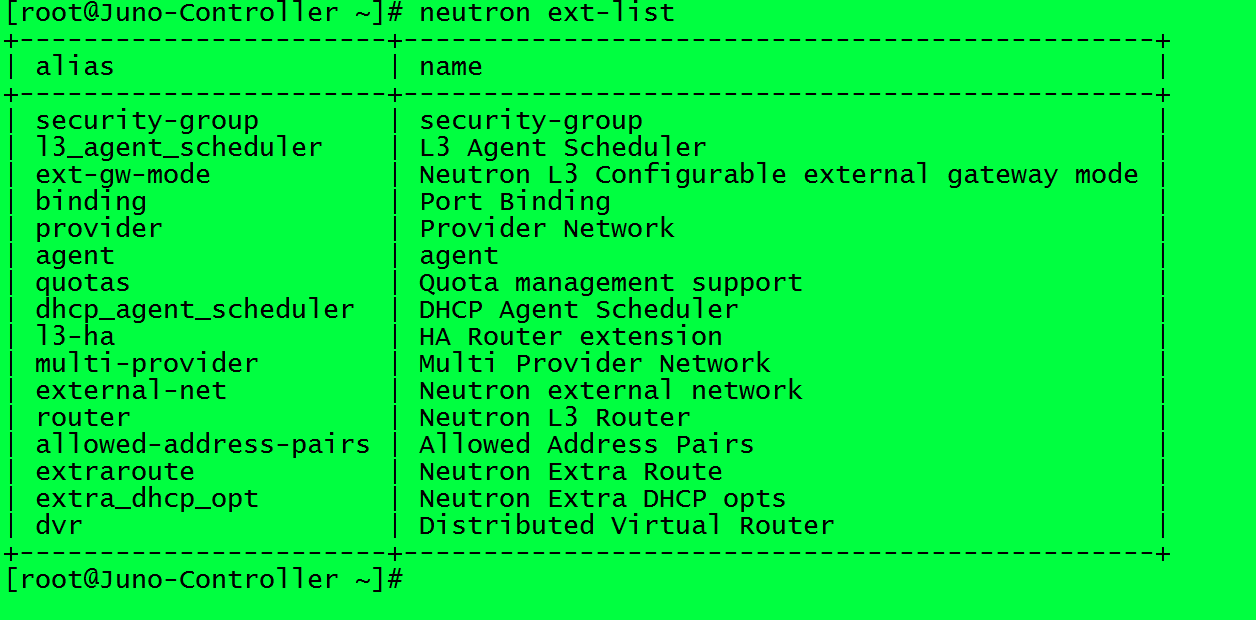
su -s /bin/sh -c "neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2\_conf.ini upgrade juno" neutron

systemctl restart openstack-nova-api.service openstack-nova-scheduler.service openstack-nova-conductor.service

systemctl enable neutron-server.service

systemctl start neutron-server.service

1. verify operation



**Network node:**

1. Edit the /etc/sysctl.conf file to contain the following parameters:

net.ipv4.ip\_forward=1

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

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

sysctl -p

1. yum install openstack-neutron openstack-neutron-ml2 openstack-neutron-openvswitch

Edit the /etc/neutron/neutron.conf file and complete the following actions:

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = cisco123

Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file and complete the following actions:

[ml2]

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_flat]

flat\_networks = external

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

[ovs]

local\_ip = 192.168.10.2

enable\_tunneling = True

bridge\_mappings = external:br-ex

[agent]

tunnel\_types = gre

Edit the /etc/neutron/l3\_agent.ini file and complete the following actions:

[DEFAULT]

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

#use\_namespaces = True

external\_network\_bridge = br-ex

#router\_delete\_namespaces = True

verbose = True

Edit the /etc/neutron/dhcp\_agent.ini file and complete the following actions:

[DEFAULT]

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

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

#use\_namespaces = True

#dhcp\_delete\_namespaces = True

verbose = True

Edit the /etc/neutron/metadata\_agent.ini file and complete the following actions:

[DEFAULT]

auth\_url = http://10.124.21.143:5000/v2.0

auth\_region = regionOne

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = cisco123

nova\_metadata\_ip = 10.124.21.143

#metadata\_proxy\_shared\_secret = cisco123

verbose = True

On the *controller* node, edit the /etc/nova/nova.conf file and complete the following action:

[neutron]

……..

#service\_metadata\_proxy = True

#metadata\_proxy\_shared\_secret = cisco123

systemctl restart openstack-nova-api.service

1. On the network node:

systemctl enable openvswitch.service

systemctl start openvswitch.service

ovs-vsctl add-br br-ex

ovs-vsctl add-port br-ex ens160

ethtool -K ens160 gro off

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

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

systemctl enable neutron-openvswitch-agent.service neutron-l3-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service neutron-ovs-cleanup.service

systemctl start neutron-openvswitch-agent.service neutron-l3-agent.service neutron-dhcp-agent.service neutron-metadata-agent.service

Notes: if you can’t start the neutron-openvswitch-agent service, pls follow tips below:

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

HWADDR=00:50:56:96:10:dc

DEFROUTE=yes

PEERDNS=yes

PEERROUTES=yes

IPV4\_FAILURE\_FATAL=no

NAME=br-ex

ONBOOT=yes

IPADDR=10.124.21.145

GATEWAY=10.124.21.1

NETMASK=255.255.255.0

NETWORK=10.124.21.0

DEVICE=br-ex

DEVICETYPE=ovs

TYPE=OVSBridge

BOOTPROTO=static

vim /etc/sysconfig/network-scripts/ifcfg-ens160

HWADDR=00:50:56:96:10:dc

NAME=ens160

ONBOOT=yes

DEVICE=ens160

TYPE=OVSPort

DEVICETYPE=ovs

OVS\_BRIDGE=br-ex

service network restart

Allow ICMP and SSH:

neutron security-group-rule-create --protocol icmp --direction ingress default

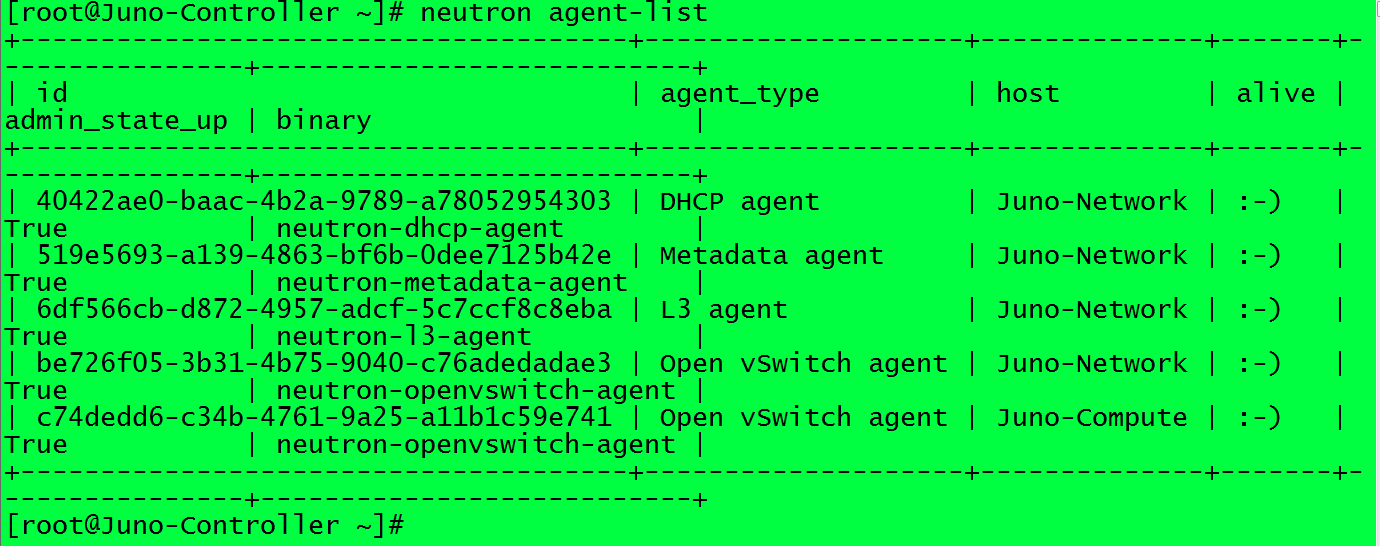
neutron security-group-rule-create --protocol tcp --port-range-min 22 --port-range-max 22 --direction ingress default

iptables -I INPUT -p tcp --dport 22 -j ACCEPT

iptables -I INPUT -p tcp --dport 22 --syn -j ACCEPT

iptables-save

1. On controller node, do verification below:



The status of alive is :-) , this means it works well.

**Compute node:**

1. Edit the /etc/sysctl.conf file to contain the following parameters:

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

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

sysctl -p

1. yum install openstack-neutron-ml2 openstack-neutron-openvswitch

Edit the /etc/neutron/neutron.conf file and complete the following actions:

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

core\_plugin = ml2

service\_plugins = router

allow\_overlapping\_ips = True

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = neutron

admin\_password = cisco123

Edit the /etc/neutron/plugins/ml2/ml2\_conf.ini file and complete the following actions:

[ml2]

type\_drivers = flat,gre

tenant\_network\_types = gre

mechanism\_drivers = openvswitch

[ml2\_type\_gre]

tunnel\_id\_ranges = 1:1000

[securitygroup]

enable\_security\_group = True

enable\_ipset = True

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

[ovs]

local\_ip = 192.168.10.1

enable\_tunneling = True

[agent]

tunnel\_types = gre

systemctl enable openvswitch.service

systemctl start openvswitch.service

Edit the /etc/nova/nova.conf file and complete the following actions:

[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

[neutron]

url = http://10.124.21.143:9696

auth\_strategy = keystone

admin\_auth\_url = http://10.124.21.143:35357/v2.0

admin\_tenant\_name = service

admin\_username = neutron

admin\_password = cisco123

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

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

systemctl restart openstack-nova-compute.service

systemctl enable neutron-openvswitch-agent.service

systemctl start neutron-openvswitch-agent.service

1. on controller node, verify the operation :

neutron agent-list

**[Create initial networks](http://docs.openstack.org/juno/install-guide/install/yum/content/neutron-initial-networks.html)**

**On controller node:**

source admin-openrc.sh

neutron net-create ext-net --router:external True --provider:physical\_network external --provider:network\_type flat

neutron subnet-create ext-net --name ext-subnet --allocation-pool start=10.124.21.150,end=10.124.21.154 --disable-dhcp --gateway 10.124.21.1 10.124.21.0/24

source demo-openrc.sh

neutron net-create demo-net

neutron subnet-create demo-net --name demo-subnet --gateway 192.168.125.1 192.168.125.0/24

neutron router-create demo-router

neutron router-interface-add demo-router demo-subnet

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

**Horizon Installation**

1. yum install openstack-dashboard httpd mod\_wsgi memcached python-memcached

Edit the /etc/openstack-dashboard/local\_settings file and complete the following actions:

OPENSTACK\_HOST = "10.124.21.143"

ALLOWED\_HOSTS = ['\*']

CACHES = {

'default': {

'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

'LOCATION': '127.0.0.1:11211',

}

}

Note: Check the indentation, Python differentiates tabs and spaces; look before CACHE and make sure everything is ok.

setsebool -P httpd\_can\_network\_connect on

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

systemctl enable httpd.service memcached.service

systemctl start httpd.service memcached.service

**Cinder Installation**

**Controller node:**

1. mysql -uroot -pcisco123

CREATE DATABASE cinder;

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder\_admin'@'%' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON cinder.\* TO 'cinder\_admin'@'localhost' IDENTIFIED BY 'cisco123';

source admin-openrc.sh

keystone user-create --name cinder --pass cisco123

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

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

keystone service-create --name cinderv2 --type volumev2 --description "OpenStack Block Storage"

keystone endpoint-create --service-id $(keystone service-list | awk '/ volume / {print $2}') --publicurl http://10.124.21.143:8776/v1/%\(tenant\_id\)s --internalurl http://10.124.21.143:8776/v1/%\(tenant\_id\)s --adminurl http://10.124.21.143:8776/v1/%\(tenant\_id\)s --region regionOne

keystone endpoint-create --service-id $(keystone service-list | awk '/ volumev2 / {print $2}') --publicurl http://10.124.21.143:8776/v2/%\(tenant\_id\)s --internalurl http://10.124.21.143:8776/v2/%\(tenant\_id\)s --adminurl http://10.124.21.143:8776/v2/%\(tenant\_id\)s --region regionOne

1. yum install openstack-cinder python-cinderclient python-oslo-db

Edit the /etc/cinder/cinder.conf file and complete the following actions:

[database]

connection = mysql://cinder\_admin:cisco123@10.124.21.143/cinder

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

my\_ip = 10.124.21.143

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = cinder

admin\_password = cisco123

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

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

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

**Compute node:**

1. systemctl enable lvm2-lvmetad.service

systemctl start lvm2-lvmetad.service

pvcreate /dev/sdb1

vgcreate cinder-volumes /dev/sdb1

vgdisplay

Edit the /etc/lvm/lvm.conf file and complete the following actions:

devices {

...

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

1. yum install openstack-cinder targetcli python-oslo-db MySQL-python

Edit the /etc/cinder/cinder.conf file and complete the following actions:

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

my\_ip = 10.124.21.144

verbose = True

glance\_host = 10.124.21.143

iscsi\_helper = lioadm

[database]

connection = mysql://cinder\_admin:cisco123@10.124.21.143/cinder

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

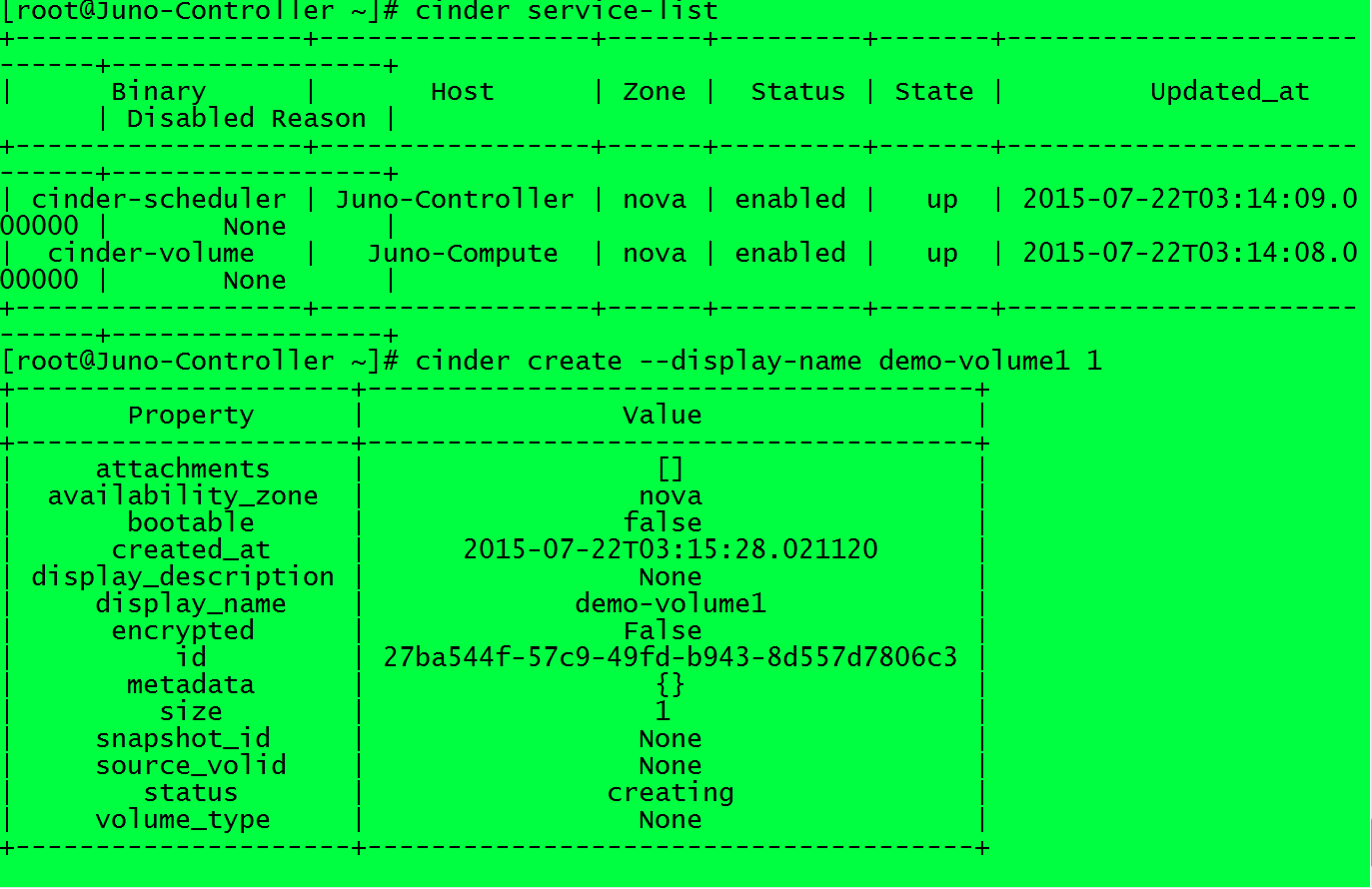
admin\_user = cinder

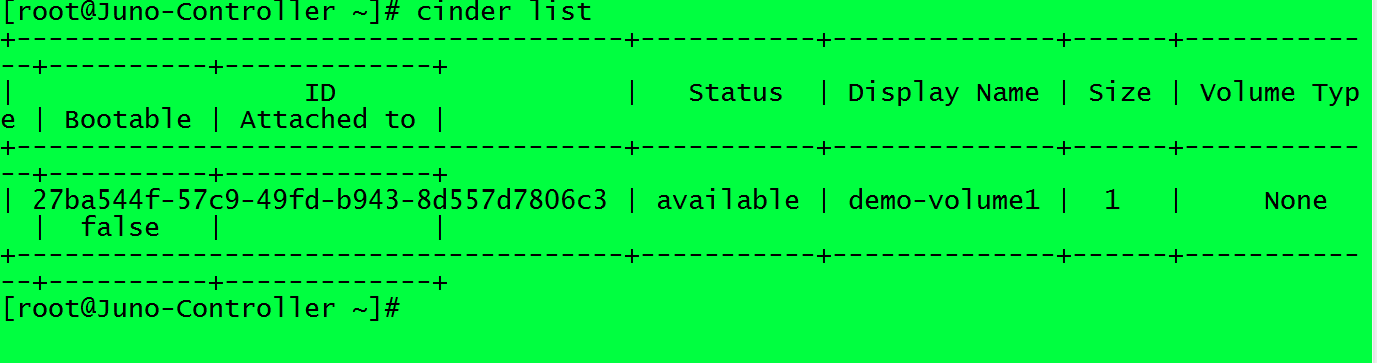
admin\_password = cisco123

systemctl enable openstack-cinder-volume.service target.service

systemctl start openstack-cinder-volume.service target.service

1. on controller node, do verification below:





**Swift Installation**

**Controller node:**

1. keystone user-create --name swift --pass cisco123

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

keystone service-create --name swift --type object-store --description "OpenStack Object Storage"

keystone endpoint-create --service-id $(keystone service-list | awk '/ object-store / {print $2}') --publicurl 'http://10.124.21.143:8080/v1/AUTH\_%(tenant\_id)s' --internalurl 'http://10.124.21.143:8080/v1/AUTH\_%(tenant\_id)s' --adminurl http://10.124.21.143:8080 --region regionOne

1. yum install openstack-swift-proxy python-swiftclient python-keystonemiddleware memcached python-keystone-auth-token

curl -o /etc/swift/proxy-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/proxy-server.conf-sample>

Edit the /etc/swift/proxy-server.conf file and complete the following actions:

[DEFAULT]

bind\_port = 8080

swift\_dir = /etc/swift

user = swift

[pipeline:main]

pipeline = authtoken cache healthcheck keystoneauth proxy-logging proxy-server

[app:proxy-server]

use = egg:swift#proxy

allow\_account\_management = true

account\_autocreate = true

[filter:keystoneauth]

use = egg:swift#keystoneauth

operator\_roles = admin,\_member\_

[filter:authtoken]

paste.filter\_factory = keystonemiddleware.auth\_token:filter\_factory

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = swift

admin\_password = cisco123

delay\_auth\_decision = true

[filter:cache]

memcache\_servers = 127.0.0.1:11211

**Compute node:**

1. mkfs.xfs /dev/sdb1

mkfs.xfs /dev/sdc1

mkdir -p /srv/node/sdb1

mkdir -p /srv/node/sdc1

Edit the /etc/fstab file and add the following to it:

/dev/sdb1 /srv/node/sdb1 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

/dev/sdc1 /srv/node/sdc1 xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

mount /srv/node/sdb1

mount /srv/node/sdc1

Edit the /etc/rsyncd.conf file and add the following to it:

uid = swift

gid = swift

log file = /var/log/rsyncd.log

pid file = /var/run/rsyncd.pid

address = 10.124.21.144

[account]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/account.lock

[container]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/container.lock

[object]

max connections = 2

path = /srv/node/

read only = false

lock file = /var/lock/object.lock

systemctl enable rsyncd.service

systemctl start rsyncd.service

1. yum install openstack-swift-account openstack-swift-container openstack-swift-object

curl -o /etc/swift/account-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/account-server.conf-sample>

curl -o /etc/swift/container-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/container-server.conf-sample>

curl -o /etc/swift/object-server.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/object-server.conf-sample>

Edit the /etc/swift/account-server.conf file and complete the following actions:

[DEFAULT]

bind\_ip = 10.124.21.144

bind\_port = 6002

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon account-server

[filter:recon]

recon\_cache\_path = /var/cache/swift

Edit the /etc/swift/container-server.conf file and complete the following actions:

[DEFAULT]

bind\_ip = 10.124.21.144

bind\_port = 6001

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon container-server

[filter:recon]

recon\_cache\_path = /var/cache/swift

Edit the /etc/swift/object-server.conf file and complete the following actions:

[DEFAULT]

bind\_ip = 10.124.21.144

bind\_port = 6000

user = swift

swift\_dir = /etc/swift

devices = /srv/node

[pipeline:main]

pipeline = healthcheck recon object-server

[filter:recon]

recon\_cache\_path = /var/cache/swift

chown -R swift:swift /srv/node

mkdir -p /var/cache/swift

chown -R swift:swift /var/cache/swift

1. Create initial rings

Controller node:

cd /etc/swift/

swift-ring-builder account.builder create 10 3 1

swift-ring-builder account.builder add r1z1-10.124.21.144:6002/sdb1 100

swift-ring-builder account.builder add r1z1-10.124.21.144:6002/sdc1 100

swift-ring-builder account.builder

swift-ring-builder account.builder rebalance

swift-ring-builder container.builder create 10 3 1

swift-ring-builder container.builder add r1z1-10.124.21.144:6001/sdb1 100

swift-ring-builder container.builder add r1z1-10.124.21.144:6001/sdc1 100

swift-ring-builder container.builder

swift-ring-builder container.builder rebalance

swift-ring-builder object.builder create 10 3 1

swift-ring-builder object.builder add r1z1-10.124.21.144:6000/sdb1 100

swift-ring-builder object.builder add r1z1-10.124.21.144:6000/sdc1 100

swift-ring-builder object.builder

swift-ring-builder object.builder rebalance

Copy the account.ring.gz, container.ring.gz, and object.ring.gz files to the /etc/swift directory on each storage node and any additional nodes running the proxy

service.

scp /etc/swift/\*.gz [root@10.124.21.144:/etc/swift/](mailto:root@10.124.21.144:/etc/swift/)

on controller node and compute node:

curl -o /etc/swift/swift.conf <https://raw.githubusercontent.com/openstack/swift/stable/juno/etc/swift.conf-sample>

Edit the /etc/swift/swift.conf file and complete the following actions:

[swift-hash]

swift\_hash\_path\_suffix = fLIbertYgibbitZ

swift\_hash\_path\_prefix = xrfuniounenqjnw

[storage-policy:0]

name = Policy-0

default = yes

chown -R swift:swift /etc/swift

on controller node:

systemctl enable openstack-swift-proxy.service memcached.service

systemctl start openstack-swift-proxy.service memcached.service

on compute node;

systemctl enable openstack-swift-account.service openstack-swift-account-auditor.service openstack-swift-account-reaper.service openstack-swift-account-replicator.service

systemctl start openstack-swift-account.service openstack-swift-account-auditor.service openstack-swift-account-reaper.service openstack-swift-account-replicator.service

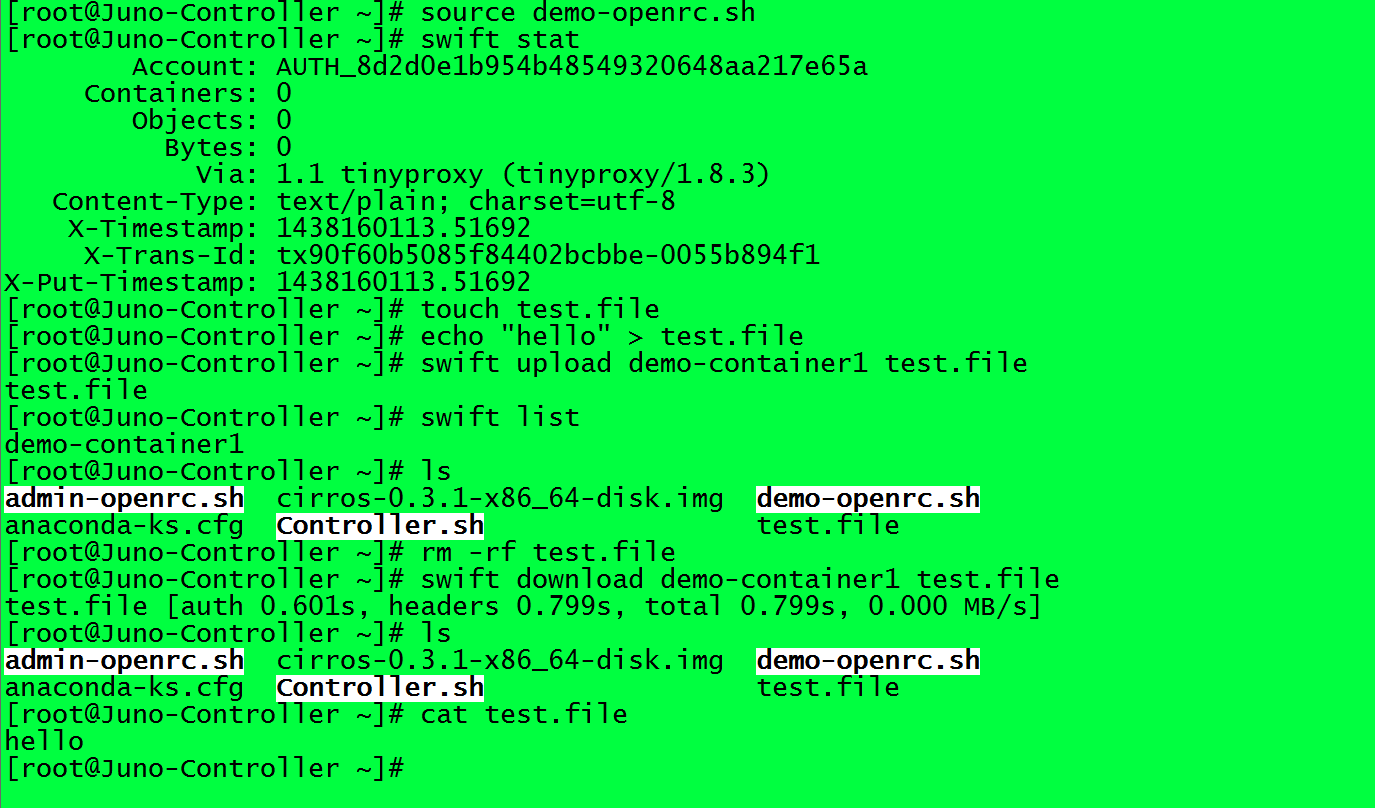
systemctl enable openstack-swift-container.service openstack-swift-container-auditor.service openstack-swift-container-replicator.service openstack-swift-container-updater.service

systemctl start openstack-swift-container.service openstack-swift-container-auditor.service openstack-swift-container-replicator.service openstack-swift-container-updater.service

systemctl enable openstack-swift-object.service openstack-swift-object-auditor.service openstack-swift-object-replicator.service openstack-swift-object-updater.service

systemctl start openstack-swift-object.service openstack-swift-object-auditor.service openstack-swift-object-replicator.service openstack-swift-object-updater.service

1. verify operation



**Heat Installation**

**Controller node:**

1. mysql -uroot -pcisco123

CREATE DATABASE heat;

GRANT ALL PRIVILEGES ON heat.\* TO 'heat\_admin'@'localhost' IDENTIFIED BY 'cisco123';

GRANT ALL PRIVILEGES ON heat.\* TO 'heat\_admin'@'%' IDENTIFIED BY 'cisco123';

keystone user-create --name heat --pass cisco123

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

keystone role-create --name heat\_stack\_owner

keystone user-role-add --user demo --tenant demo --role heat\_stack\_owner

keystone role-create --name heat\_stack\_user

keystone service-create --name heat --type orchestration --description "Orchestration"

keystone service-create --name heat-cfn --type cloudformation --description "Orchestration"

keystone endpoint-create --service-id $(keystone service-list | awk '/ orchestration / {print $2}') --publicurl http://10.124.21.143:8004/v1/%\(tenant\_id\)s --internalurl http://10.124.21.143:8004/v1/%\(tenant\_id\)s --internalurl http://10.124.21.143:8004/v1/%\(tenant\_id\)s --region regionOne

keystone endpoint-create --service-id $(keystone service-list | awk '/ cloudformation / {print $2}') --publicurl http://10.124.21.143:8000/v1 --publicurl http://10.124.21.143:8000/v1 --adminurl http://10.124.21.143:8000/v1 --region regionOne

1. yum install openstack-heat-api openstack-heat-api-cfn openstack-heat-engine python-heatclient

Edit the /etc/heat/heat.conf file and complete the following actions:

[database]

connection = mysql://heat\_admin:cisco123@10.124.21.143/heat

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

heat\_metadata\_server\_url = http://10.124.21.143:8000

heat\_waitcondition\_server\_url = <http://10.124.21.143:8000/v1/waitcondition>

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = heat

admin\_password = cisco123

[ec2authtoken]

auth\_uri = <http://10.124.21.143:5000/v2.0>

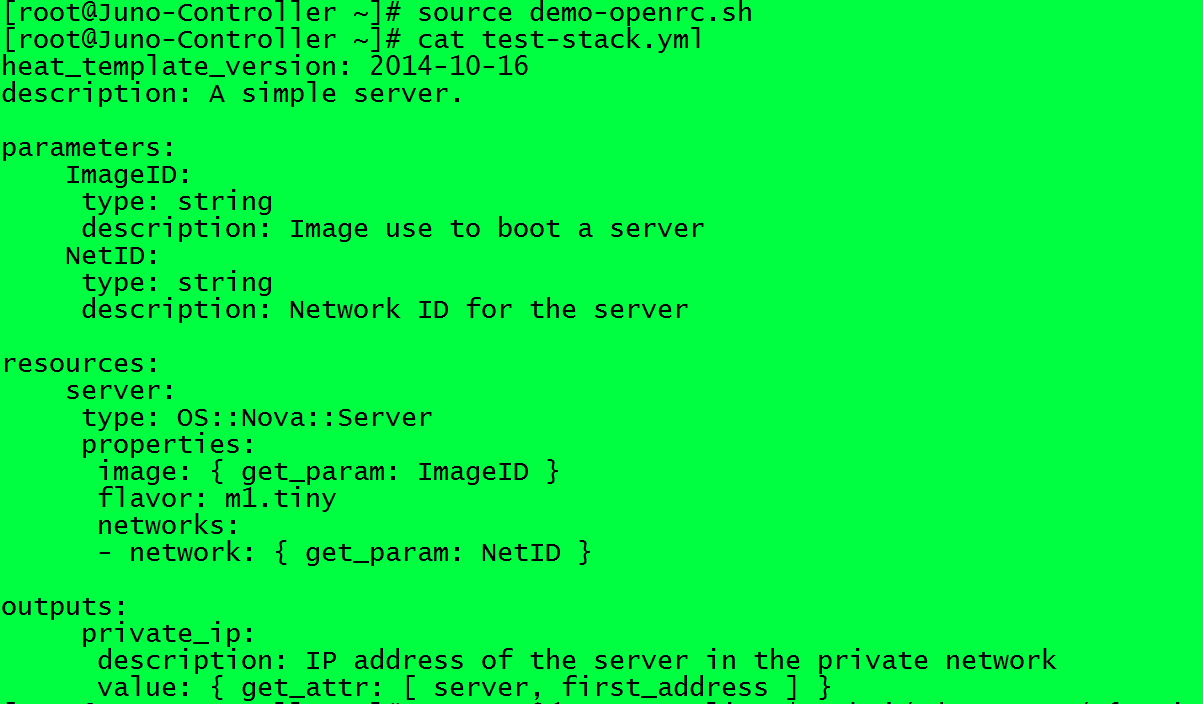
su -s /bin/sh -c "heat-manage db\_sync" heat

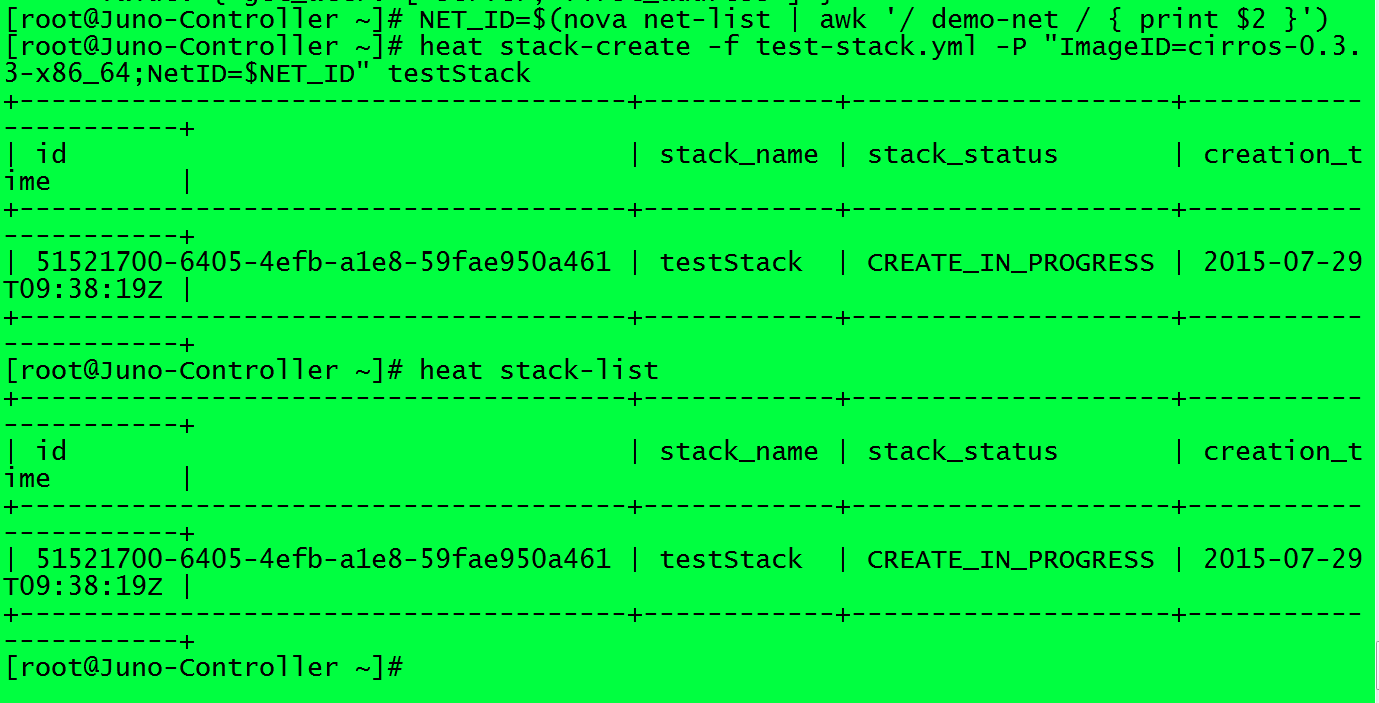
systemctl enable openstack-heat-api.service openstack-heat-api-cfn.service openstack-heat-engine.service

systemctl start openstack-heat-api.service openstack-heat-api-cfn.service openstack-heat-engine.service

1. verify operation

source demo-openrc.sh





**Ceilometer Installation**

**Controller node:**

1. yum install mongodb-server mongodb

Edit the /etc/mongod.conf file and complete the following actions:

bind\_ip = 10.124.21.143

smallfiles = true

systemctl enable mongod.service

systemctl start mongod.service

mongo --host 10.124.21.143 --eval 'db = db.getSiblingDB("ceilometer");db.addUser({user: "ceilometer",pwd: "cisco123",roles: [ "readWrite", "dbAdmin" ]})'

source admin-openrc.sh

keystone user-create --name ceilometer --pass cisco123

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

keystone service-create --name ceilometer --type metering --description "Telemetry"

keystone endpoint-create --service-id $(keystone service-list | awk '/ metering / {print $2}') --publicurl http://10.124.21.143:8777 --internalurl http://10.124.21.143:8777 --adminurl http://10.124.21.143:8777 --region regionOne

yum install openstack-ceilometer-api openstack-ceilometer-collector openstack-ceilometer-notification openstack-ceilometer-central openstack-ceilometer-alarm python-ceilometerclient

1. Edit the /etc/ceilometer/ceilometer.conf file and complete the following actions:

[database]

connection = mongodb://ceilometer:cisco123@10.124.21.143:27017/ceilometer

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

auth\_strategy = keystone

verbose = True

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = ceilometer

admin\_password = cisco123

[service\_credentials]

os\_auth\_url = http://10.124.21.143:5000/v2.0

os\_username = ceilometer

os\_tenant\_name = service

os\_password = cisco123

[publisher]

metering\_secret = cisco123

systemctl enable openstack-ceilometer-api.service openstack-ceilometer-notification.service openstack-ceilometer-central.service openstack-ceilometer-collector.service openstack-ceilometer-alarm-evaluator.service openstack-ceilometer-alarm-notifier.service

systemctl start openstack-ceilometer-api.service openstack-ceilometer-notification.service openstack-ceilometer-central.service openstack-ceilometer-collector.service openstack-ceilometer-alarm-evaluator.service openstack-ceilometer-alarm-notifier.service

**Configure the compute service:**

1. yum install openstack-ceilometer-compute python-ceilometerclient python-pecan

Edit the /etc/ceilometer/ceilometer.conf file and complete the following actions:

[publisher]

metering\_secret = cisco123

[DEFAULT]

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

[keystone\_authtoken]

auth\_uri = http://10.124.21.143:5000/v2.0

identity\_uri = http://10.124.21.143:35357

admin\_tenant\_name = service

admin\_user = ceilometer

admin\_password = cisco123

[service\_credentials]

os\_auth\_url = http://10.124.21.143:5000/v2.0

os\_username = ceilometer

os\_tenant\_name = service

os\_password = cisco123

os\_endpoint\_type = internalURL

os\_region\_name = regionOne

Edit the /etc/nova/nova.conf file and configure notifications in the [DEFAULT]section:

[DEFAULT]

instance\_usage\_audit = True

instance\_usage\_audit\_period = hour

notify\_on\_state\_change = vm\_and\_task\_state

notification\_driver = messagingv2

systemctl enable openstack-ceilometer-compute.service

systemctl start openstack-ceilometer-compute.service

systemctl restart openstack-nova-compute.service

**Configure the Image Service:**

1. Edit the /etc/glance/glance-api.conf and /etc/glance/glance-registry.conf files and complete the following actions:

[DEFAULT]

#notification\_driver = noop

notification\_driver = messagingv2

rpc\_backend = rabbit

rabbit\_host = 10.124.21.143

rabbit\_password = cisco123

systemctl restart openstack-glance-api.service openstack-glance-registry.service

**Configure the block storage service:**

1. Edit the /etc/cinder/cinder.conf file and complete the following actions:

[DEFAULT]

control\_exchange = cinder

notification\_driver = messagingv2

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

systemctl restart openstack-cinder-volume.service (on compute node)

**Configure the object storage service:**

1. source admin-openrc.sh

keystone role-create --name ResellerAdmin

keystone user-role-add --tenant service --user ceilometer --role 42c4bbb83a124c16b1fe5e4a17a554a6

Edit the /etc/swift/proxy-server.conf file and complete the following actions:

[filter:keystoneauth]

use = egg:swift#keystoneauth

operator\_roles = admin,\_member\_,ResellerAdmin

[pipeline:main]

pipeline = authtoken cache healthcheck keystoneauth proxy-logging ceilometer proxy-server

[filter:ceilometer]

use = egg:ceilometer#swift

log\_level = WARN

usermod -a -G ceilometer swift

chmod -R 777 /var/log/ceilometer/

systemctl restart openstack-swift-proxy.service

## Verify the Telemetry installation

## source admin-openrc.sh

## ceilometer meter-list

## glance image-download "cirros-0.3.3-x86\_64" > cirros.img

## ceilometer meter-list | grep "image"

## ceilometer statistics -m image.dowload -p 60